

MARTIN-CR-65-5, Vol. II END

Copy No. 15

CONTRACT NAS-8-11415 - 100%

3 OPTIMIZATION OF HYDRAULIC THRUST VECTOR
CONTROL SYSTEMS FOR LAUNCH VEHICLES

9 JANUARY, 1965 11:30

VOLUME II: 2A
COMPUTER PROGRAM DESCRIPTION,
INSTRUCTIONS, AND LISTING 9

THE MARTIN-MARIETTA CORPORATION
DENVER DIVISION
POST OFFICE BOX 719
DENVER 1, COLORADO

FOREWORD

This document is submitted in accordance with Contract NAS
8-11415. The report is in three volumes as follows:

Volume I - Study Program Description and Results
for Saturn F1 and J2

Volume II - Computer Program Description, Instructions
and Listing.

Volume III - Derivation of Equations

CONTENTS

| | |
|---|-----|
| FOREWORD | i |
| CONTENTS | ii |
| LIST OF FIGURES | iii |
| LIST OF TABLES | iv |
| I. INTRODUCTION | 1 |
| II. COMPUTER PROGRAM DISCUSSION | 4 |
| A. General Program Discussion | 4 |
| 1. Supervisor Program | 4 |
| 2. Disk Utility Program | 5 |
| 3. Fortran II-D Program | 5 |
| B. Discussion of Individual Programs | 6 |
| C. Initial Data Input | 7 |
| D. Program Output and Results | 28 |
| 1. Final Results | 28 |
| 2. Intermediate Component Results | 31 |
| E. General Instructions for Running Program | 32 |
| F. Listing of Component Computer Program | 40 |

LIST OF FIGURES

| | | |
|-----|----------------------------------|----|
| I. | ENGINE DIMENSION COMPUTER INPUTS | 19 |
| II. | EXAMPLE I.B.M. DATA CARD | 20 |

LIST OF TABLES

| | | |
|--------|---|----|
| I. | INITIAL PARAMETER DATA CARDS | 18 |
| II. | INPUT DATA FOR SATURN S-IC COMPUTER RUNS | 22 |
| III. | INPUT DATA FOR SATURN S-II COMPUTER RUNS | 24 |
| IV. | INPUT DATA FOR SATURN S-IVB COMPUTER RUNS | 26 |
| V. | O-RING WEIGHT SUBROUTINE COMPUTER PROGRAM LISTING | 40 |
| VI. | O-RING FAILURE RATE SUBROUTINE COMPUTER PROGRAM LISTING | 41 |
| VII. | DECK NUMBER 1-ACTUATOR, COMPUTER PROGRAM LISTING | 43 |
| VIII. | DECK NUMBER 2-ACTUATOR, COMPUTER PROGRAM LISTING | 46 |
| IX. | DECK NUMBER 3-ACTUATOR, COMPUTER PROGRAM LISTING | 48 |
| X. | DECK NUMBER 4-ACTUATOR & TRUSS, COMPUTER PROGRAM LISTING | 51 |
| XI. | DECK NUMBER 5-TUBING, COMPUTER PROGRAM LISTING | 54 |
| XII. | DECK NUMBER 6-FIXED ANGLE PUMP, COMPUTER PROGRAM LISTING | 57 |
| XIII. | DECK NUMBER 7-FIXED ANGLE PUMP, COMPUTER PROGRAM LISTING | 61 |
| XIV. | DECK NUMBER 8-WOBBLE PLATE PUMP, COMPUTER PROGRAM LISTING | 65 |
| XV. | DECK NUMBER 9-INTENSIFIER, COMPUTER PROGRAM LISTING | 68 |
| XVI. | DECK NUMBER 10-FILTER, COMPUTER PROGRAM LISTING | 71 |
| XVII. | DECK NUMBER 11-RES.-ACCUM., COMPUTER PROGRAM LISTING | 73 |
| XVIII. | DECK NUMBER 12-OVERALL VEHICLE, COMPUTER PROGRAM LISTING | 76 |

I. INTRODUCTION

This report is the final technical summary report for contract NAS 8-11415. The objective of the work performed under this contract was to conduct studies for the optimization of hydraulic thrust vector control systems for launch vehicles based upon reliability, performance and weight. The systems considered were for the Saturn F1 and J2 engines. The two parameters of the hydraulic system which could be adjusted for optimization were the system pressure and the engine-actuator moment arm.

The general approach used for this study was to derive mathematical model equations which represented the system weight, reliability, etc. solve these equations (using a digital computer) for various combinations of moment arm and pressure, prepare graphs of the results, and then from the graphs determine which combination of the two system variables resulted in the best or optimal system. The selection of the optimum point required converting all characteristics to some common denominator. This was accomplished by converting each characteristic to a representative cost or dollar value to the program. The resulting cost factor included the cost of weight, failures, etc.

Using the above approach, the analysis and resulting equations were relatively complex and required solution for each data point on the graphs. It was therefore decided that the objectives of this study could best be presented by formulating a computer program of the model equations.

During the compiling of the computer program, the program was maintained as flexible as possible, allowing for variations in parameters other than pressure and moment arm. The resulting computer program developed can be used for optimization studies of any conventional launch vehicle utilizing hydraulic operated thrust vector control systems. The computer program also permits comparative studies of such alternatives as electrical versus mechanical feedback, using a new design versus using a system which has been previously designed and proven, wobble plate pumps versus fixed angle pumps and many others. Although such a computer program was not required by the contract, it is being submitted to NASA as part of the results from this study program.

The primary objective of this report is to present a description of the study including the approach and methods used, the reasons for the approach, and the conclusion and recommendations resulting from this study. In order to facilitate usage of the program results, this report is divided into three volumes.

Volume I describes the general approach used, the various system parameters considered and specific results for both the Saturn J-2 (S-II and S-IV-B stages) and F-1 (SIC Stage) engines. This volume only is required for determining the optimum operating points for these specific Saturn systems. Volume I contains all the basic results required for this contract.

Volume II describes the general computer program that was developed for studying optimization, instructions for use and a

listing of each computer program deck. This volume only is required for use of the computer program for studying optimization of any standard hydraulic thrust vector control systems.

Volume III lists the derivation of the equations used in the computer program and is required only for verification of the computer program or making changes to the computer program.

II. COMPUTER PROGRAM DISCUSSION

A. General Program Discussion

The computer program has been designed around an I.B.M. 1620, Mark II computer. The base machine's capabilities are augmented by adding an additional forty thousand core storage unit and an I.B.M. 1311 Disk Storage Drive. This combined unit then utilizes the Monitor I Programming System which coordinates computer activity by providing a communication region for independent programming systems, and by transferring control between them. The Monitor I System used utilizes the large storage capacity of the 1311 Disk Storage Drive. Thus it is possible to assembly, assembly and execute, compile, compile and execute, and execute programs stored in disk storage. By using this system, the O-ring subroutine programs can be stored in disk storage without the necessity of assigning actual storage areas. These subroutines are easily referred to by code. The code used was OSWE for the "O" ring weight subroutine, and OSFR for the "O" ring reliability subroutine.

The Monitor I system is comprised of four separate parts; 1) Supervisor Program, 2) Disk utility program, 3) Fortran II-D, and 4) SPS II-D. Only the first three portions of the system are used and they are described below.

1. Supervisor Program

The Supervisor Program performs the control functions and input/output functions for the system. The Fortran II-D compiler, in and under the control of the Supervisor Program,

will change the initial program into machine language. This program takes the form of two cards at the start of the component programs. The first card is a "JOB" card which lists the name of the component program (Actuator, truss, pump, etc.). The second Supervisor Program card will be "FORX" which causes the computer to change the component program from Fortran language into useable machine language and work the program to obtain the desired results.

2. Disk Utility Program

This program has only one function in this study which is to store the weight and reliability "O" ring subroutines on permanent disk storage. Only one card will be needed which will be "*LDISK" and is the first card for each "O" ring subroutines. This will place the subroutines into permanent storage.

3. Fortran II-D Program

This program is the main body of the individual component programs which takes the form of Arithmetic statements. These programs specify the mathematical calculations to be performed and includes the control statements which govern the sequence of operation and the Input/Output statements. The input/output statements read data into the program and print or punch the finished program results.

B. Discussion of Individual Component Programs

Each of the component decks are written in Fortran II-D language. In each of the programs, only pertinent dimensions and information is calculated independently with the rest being combined into large equations. This approach is used in order to conserve computer memory space. Two of the component programs exceeded the computer unit storage capacity and therefore were split into sections with the required information carried on cards between sections. The actuator program was split into four separate decks, and the fixed angle pump split into two separate parts. No output information will be available from these two programs until the last portion of the programs are performed.

The study has been designed to run the programs in a set sequence with all required inputs (moment arm, pump speeds, tubing lengths, etc.) inserted, in data form, into the first deck of the actuator program. From this point on, the operation will be automatic with the operator only having to insert the next program into the computer and placing the punched output data from the previous program as input data to the deck. This will be further explained in Section E, General Instructions for Running the Program. The individual program decks are not designed to be used independently unless slight changes are made to them. An individual who desires to investigate a particular component would first have to acquaint himself with the fundamental equations (Volume III), basic component programs (end of Volume II), and Fortran II-D terminology.

The complete program must be performed in the following sequence. First, the "O" ring weight and reliability subroutines must be placed on permanent disk storage. Then the initial required inputs must be punched on cards and run with the first deck of the actuator-truss program. The rest of the program is then run in the following sequence:
1) Second actuator-truss deck, 2) Third actuator-truss deck, 3) Fourth actuator-truss deck, 4) Tubing deck, 5) First fixed angle pump deck, 6) Second fixed angle pump deck, 7) Inline wobble plate pump deck, 8) Hydraulic intensifier deck, 9) Filter deck, 10) Reservoir-accumulator deck, 11) Complete vehicle conversion deck. For each deck, the input data will be the punched data card output from the previous deck.

C. Initial Data Input

The initial system parameters (moment arm, system pressure, pump speeds, type of actuator feedback to be used, etc.) will be punched into thirteen separate I.B.M. cards and used as input data to the first deck of the actuator-truss program. Each of the thirteen data cards will have up to seven separate parameters punched onto it in the Fortran II-D floating-point constant form. Each of the seven separate parameters must be located within a ten space area on the I.B.M. card, giving a total used area of seventy spaces out of eighty available spaces per card. A parameter may be located anywhere within the ten space area and will be in the form of actual numbers with decimal points (319., 4.57, .00043, etc.). The thirteen cards and their individual parameters are listed in Table I shown below in the order in which they must appear. The parameters are given in code

and this code is explained below. The parameters must be in the units as shown.

Data Card #1

| | | |
|-------|---|---|
| TORQ | - | required maximum torque (stall) to engine (inch-pounds) |
| VELS | - | required maximum angular velocity (based on loaded actuator velocity) (Radians/Second) |
| TRAAC | - | required total operating angular travel (Radians) (does not include snubbing) |
| AKVEL | - | required open loop gain of actuator (1/Second) |
| EINT | - | engine inertia (inch-pound-second ²) |
| ANUMB | - | number of actuator per main pump |
| AVPR | - | ratio of actual valve flow rate to required valve flow rate (loaded actuator) Note: This parameter is included for the case where a miniture servovalve could be used but a larger valve along with a flow limiter is actually employed. |

Data Card #2

| | | |
|------|---|---|
| PREI | - | lowest system pressure to be investigated (pounds/inch ²) |
| DPRE | - | pressure increment to be used during program run (pounds/inch ²) |
| PREM | - | maximum system pressure to be investigated (pounds/inch ²) |

AMOM - shortest moment arm to be investigated (inches)
 DMOM - moment arm increment to be used during program run (inches)
 AMAX - longest moment arm to be investigated (inches)
 VPRON - vehicle reference number (guide number for system being investigated) Note: Any number may be used here for future reference to computer run.

Data Card #3

AAAA1 - Is actuator pressure feedback or derivative pressure feedback valve used? (If answer is yes, set AAAA1 = 1.0, if no, set to 0.0)
 AAAA2 - Is actuator mechanical feedback used? (If answer is yes, set AAAA2 = 1.0, if no, set to 0.0)
 AAAA3 - Is actuator rod end housing used for bearing surface? (If yes, set AAAA3 = 1.0, if no, set to 0.0)
 AAAA4 - Is actuator derivative pressure feedback used? (If yes, set AAAA4 = 1.0, if no, set to 0.0)
 AAAA5 - Is actuator mechanical feedback used? (If yes, set AAAA5 = 1.0, if no, set to 0.0)
 AAAA6 - Is actuator static load error washout used? (If yes, set AAAA6 = 1.0, if no, set to 0.0)
 AAAA7 - Is actuator flow limiter used? (If yes, set AAAA7 = 1.0, if no, set to 0.0)

Data Card #4

AAAA8 - Are actuator snubbers used? (If yes, set AAAA8 = 1.0, if no set to 0.0)

AAAA9 - Is the actuator a new design? (If yes, set
 AAAA9 = 1.0, if no, set to 0.0)

AAAL0 - Does the actuator require qualification? (If
 yes, set AAAL0 = 1.0, if no, set to 0.0)

AIPA1 - Is the actuator direct current position instru-
 mentation used? (If yes, set AIPA1 = 1.0, if
 no, set to 0.0)

AIPA2 - Is actuator direct current feedback used? (If
 yes, set AIPA2 = 1.0, if no, set to 0.0)

AIPA3 - Are actuator position switches used? (If yes,
 set AIPA3 = 1.0, if no, set to 0.0)

AIPA4 - Is potentiometer body required? (If yes, set
 AIPA4 = 1.0, if no, set to 0.0)

Data Card #5

XMDC - truss dimension (inches) See Figure 1

XMDD - truss dimension (inches) See Figure 1

XMDB - truss dimension (inches) See Figure 1

XMDA - truss dimension (inches) See Figure 1

XMDE - truss dimension (inches) See Figure 1

XXXX1 - Is truss a new design? (If yes, Set XXXX1
 = 1.0, if no, set to 0.0)

AKENG - Fixed spring rate of engine bell.
 (pounds/inch)

Data Card #6

- S1 - tube length from airborne pump to manifold
(inches)
- S2 - tube length from ground checkout pump to
manifold (inches)
- S3 - tube length from manifold to bottom of truss
(inches)
- S4 - tube length from truss clevis to actuator
(inches)
- XXXX2 - is tubing a new design? (If yes, set XXXX2 =
1.0, if no, set to 0.0)
- PPPP9 - ratio of the required pump flow (for either
wobble plate pump or intensifier) to system
flow rate (unloaded actuators)

The seventh parameter of data card #6 is left blank.

Data Card #7

- PPPP1 - ratio of the maximum required pump flow rate
(for fixed angle pump) to maximum system flow
rate (unloaded actuators)
- ANGL1 - angle of fixed angle pump (radians)
- ANGL2 - angle of wobble plate pump (radians)
- PUMS1 - fixed angle pump speed (revolutions per second)
- PUMS2 - wobble plate pump speed (revolutions per second)
- S5 - Is compensator used in fixed angle pump? (If
yes, set S5 = 1.0, if no, set to 0.0)
- S6 - Is compensator used in wobble plate pump? (If
yes, set S6 = 1.0, if no set to 0.0)

Data Card #8

S7 - number of fixed angle pumps per hydraulic system
 S8 - number of wobble plate pumps per hydraulic system
 S9 - number of intensifiers per hydraulic system
 PPPP2 - Is fixed angle pump a new design? (If yes, set PPPP2 = 1.0, if no, set to 0.0)
 PPPP3 - Does fixed angle pump require qualification? (If yes, set PPPP3 = 1.0, if no, set to 0.0)
 PPPP4 - Is wobble plate pump a new design? (If yes, set PPPP4 = 1.0, if no, set to 0.0)
 PPPP5 - Does wobble plate pump require qualification? (If yes, set PPPP5 = 1.0, if no, set to 0.0)

Data Card #9

PPPP6 - Is intensifier a new design? (If yes, set PPPP6 = 1.0, if no, set to 0.0)
 PPPP7 - Does intensifier require qualification? (If yes, set PPPP7 = 1.0, if no, set to 0.0)
 FFFF1 - number of filters per hydraulic system
 FFFF2 - ratio of required filter flow to the maximum actuator flow (unloaded actuators)
 FFFF3 - Is filter a new design? (If yes, set FFFF3 = 1.0, if no, set to 0.0)
 FFFF4 - Does filter require qualification? (If yes, set FFFF4 = 1.0, if no set to 0.0)
 PPPP8 - Which airborne pump used for ground checkout? (No airborne pump used set PPPP8 = 0.0, fixed angle pump used set PPPP8 = 1.0, wobble plate pump used set PPPP8 = 2.0)

Data Card #10

SSS2 - ratio of return pressure to system pressure if return pressure is a function of system pressure.

SSS3 - return pressure as a fixed actual value if the return pressure is to be held constant (pounds/inch²)

SSS1 - Is accumulator used? (If yes, set SSS1 = 1.0, if no, set to 0.0)

RSPA1 - Is direct current position instrumentation used in the reservoir? (If yes, set RSPA1 = 1.0, if no, set to 0.0)

RSPA2 - Is position switch used in reservoir? (If yes, set RSPA2 = 1.0, if no, set to 0.0)

RSPA3 - Is potentiometer body integral part of the reservoir? (If yes, set RSPA3 = 1.0, if no, set to 0.0)

TOILW - Hydraulic fluid density used in system. (pounds per cubic inch)

Data Card #11

RRRR1 - ratio of total volume of fluid supplied by the accumulator to the total volume of fluid consumed by all actuators when traveling full stroke.

RRRR3 - Is reservoir and/or accumulator a new design? (If yes, set RRRR3 = 1.0, if no, set to 0.0)

RRRR4 - Does reservoir and/or accumulator require qualification? (if yes, set RRRR4 = 1.0, if no, set to 0.0)

QQQQ1 - Is quick disconnect a new design? (If yes, set QQQQ1 = 1.0, if no, set to 0.0)

QQQQ2 - Does quick disconnect require qualification? (If yes, set QQQQ2 = 1.0, if no, set to 0.0)

QQQQ3 - ratio of quick disconnect rated flow to the maximum system flow rate (unloaded actuators)

VNAFQ - required actuator system natural frequency with all springs included (radians/second)

Data Card #12

VHSFO - total system operating time during flight (hours)

VFLRC - Cost of generic failure during flight. For the cost of failures, the generic failure rate for the hydraulic system is first multiplied by the number of flight systems to be used in the launch vehicle program and then by the total flight time for one system. This yields a result which is the total number of flight generic failures for the hydraulic system that can be expected during the complete program. In order to convert this value to a dollar value for the program, it must be multiplied by the application factor K_a (generally equal to 1.0), and the operating mode factor K_{op} (generally

between 800 and 2000 depending on the environment and stage), to arrive at the failures per program and finally by the cost of a single failure in the hydraulic system during flight. Since the cost of a failure, K_a and K_{op} can vary between vehicles, these three factors have been combined into a single term (VFLRC) which is the cost in dollars of a generic flight failure.

| | | |
|-------|---|--|
| VPNUB | - | total number of launch vehicles within the program |
| VHYSB | - | total number of independent hydraulic systems to be used for the particular stage of the vehicle. |
| VWCST | - | the cost of one pound of weight for a particular stage being investigated. (dollars per pound) |
| VDEVL | - | total time allowed to develop the complete hydraulic system. Equal to the total time from contract go-ahead until a qualified system is delivered. (weeks) |
| VPEND | - | dollar penalty per week for delays in development time for the complete system. (dollars per week) |

Data Card #13

| | | |
|-------|---|---|
| VREPR | - | average ratio of component repair cost to initial component cost. |
| VLIFP | - | required life of a single pump. Includes total running time - ground checkout, flight, etc. (hours) |

VLIFA - required life of a single actuator includes total running time - ground checkout, flight, etc. (hours)

VCYCA - required life of a single actuator in total number of cycles - ground checkout, flight, etc. (cycles)

VOPER - required time for average hydraulic test on the system under consideration. (hours)

VTEST - total number of tests required for a single hydraulic system under investigation.

VTCST - total cost of average test performed on a single hydraulic system under investigation. (dollars per test)

This completes the listing of the required system data parameters to be punched into the thirteen initial data cards.

It is important to remember that each parameter number must have a decimal point, must fall within the ten space area assigned (including decimal point), and must be punched into the thirteen data cards in the sequence listed above. As an example, data card #1 parameters may be as follows:

| | | |
|-------|---|---------|
| TORQ | = | 540,000 |
| VELS | = | 0.0825 |
| TRAA | = | 0.18 |
| AKVEL | = | 18.0 |
| EINT | = | 372,000 |
| ANUMB | = | 2.0 |
| AVPR | = | 1.0 |

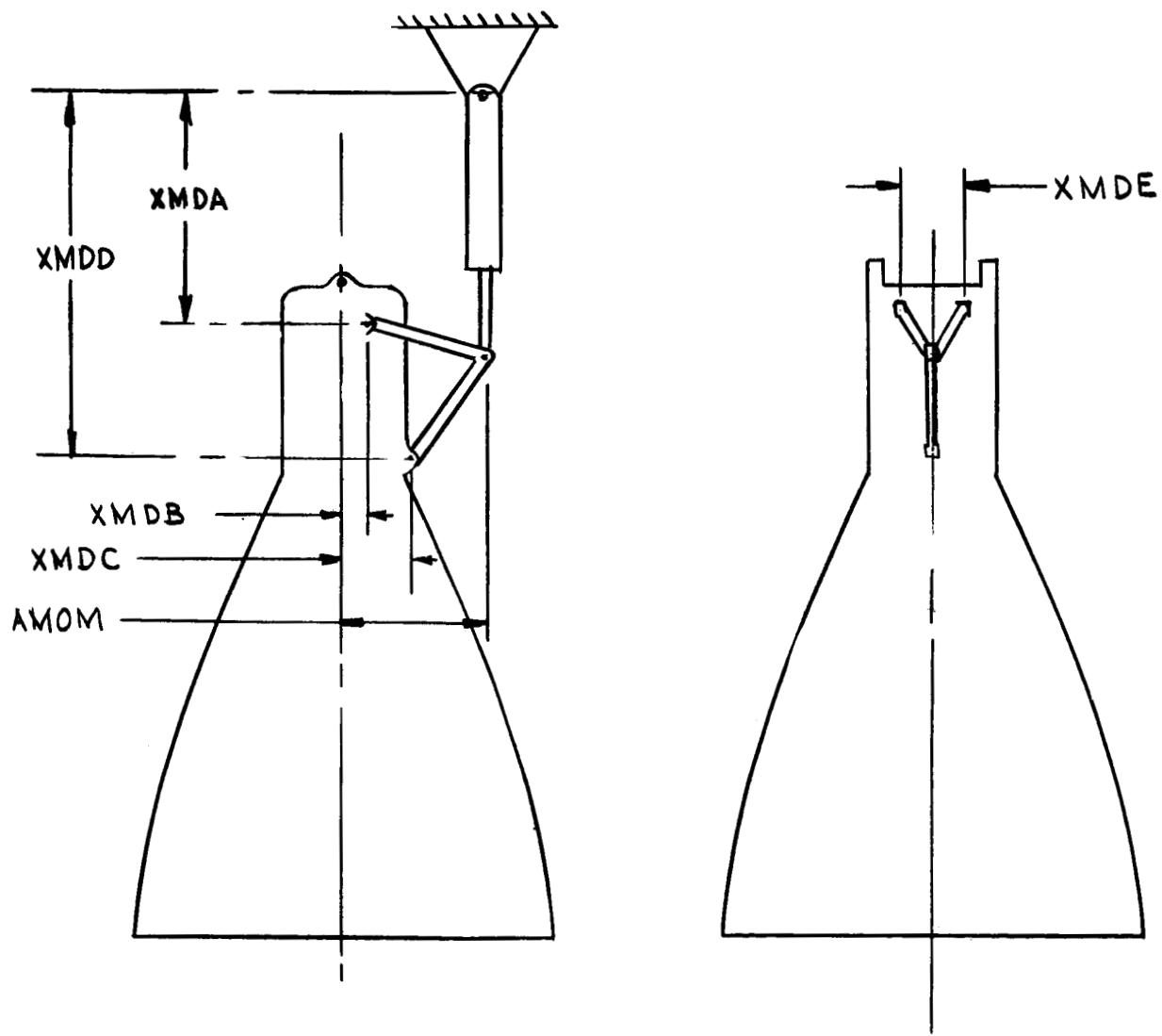
This parameter data would then appear on an I.B.M. data card as shown in Figure II and would be used on Data Card #1 in the initial data group.

TABLE I
INITIAL PARAMETER DATA CARDS

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------------------|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Input Card Number | Card Space 1-10 | Card Space 11-20 | Card Space 21-30 | Card Space 31-40 | Card Space 41-50 | Card Space 51-60 | Card Space 61-70 |
| 1 | TORQ | VELS | TRA | AKVEL | EINT | ANUMB | AVPR |
| 2 | PREI | DPRE | PREM | AMOM | DMOM | AMAX | VPRON |
| 3 | AAAA1 | AAAA2 | AAA3 | AAA4 | AAA5 | AAA6 | AAA7 |
| 4 | AAA8 | AAA9 | AAA10 | AIPA1 | AIPA2 | AIPA3 | AIPA4 |
| 5 | XMD | XMD | XMD | XMDA | XMD | XXX1 | AKENG |
| 6 | S1 | S2 | S3 | S4 | XXX2 | PPP9 | - |
| 7 | PPP1 | ANGL1 | ANGL2 | PUMS1 | PUMS2 | S5 | S6 |
| 8 | S7 | S8 | S9 | PPP2 | PPP3 | PPP4 | PPP5 |
| 9 | PPP6 | PPP7 | FFF1 | FFF2 | FFF3 | FFF4 | PPP8 |
| 10 | SSS2 | SSS3 | SSS1 | RSPA1 | RSPA2 | RSPA3 | TOILW |
| 11 | RRR1 | RRR3 | RRR4 | QQQ1 | QQQ2 | QQQ3 | VNAFQ |
| 12 | VHSFO | VFLRC | VPNUB | VHYSB | VWCST | VDEVL | VPEND |
| 13 | VREPR | VLIFP | VLIFA | VCYCA | VOPER | VTEST | VTCST |

FIGURE I

ENGINE TRUSS DIMENSION TERMINOLOGY



EXAMPLE I.B.M. DATA CARD

FIGURE II

The following tables list the computer "inputs" used for the Saturn Program Study in Volume I.

| | | | | | | |
|----------|-------------|------|---------|---------|------|---------|
| 5607000. | .0825 | .18 | 18.0 | 372000. | 2.0 | 1.0 |
| 1540. | 0.0 | 0.0 | 63.76 | 0.0 | 0.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 34.0 | 102. | 32. | 40. | 22. | 0.0 | 397600. |
| 54.0 | 30. | 84. | 36. | 0.0 | 0.0 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| .040 | 2000000000. | 20.0 | 4.0 | 60.0 | 52.0 | 12000. |
| .35 | 0.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|----------|-------------|-------|---------|---------|------|---------|
| 5607000. | .0825 | .18 | 18.0 | 372000. | 2.0 | 1.0 |
| 1500. | 500. | 4500. | 40. | 10. | 100. | 4.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 34.0 | 102. | 32. | 40. | 22. | 1.0 | 397600. |
| 54.0 | 30. | 84. | 36. | 1.0 | 1.0 | |
| 0.0 | 0.0 | .28 | 0.0 | 83.5 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1.0 | 1.0 | 1.0 | 0.2 | 1.0 | 1.0 | 0.0 |
| 0.01333 | 10.0 | 0.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 0.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 50.0 |
| .040 | 2000000000. | 20.0 | 4.0 | 60.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|----------|-------------|------|---------|---------|------|---------|
| 5607000. | .0825 | .18 | 18.0 | 372000. | 2.0 | 1.0 |
| 1540. | 0.0 | 0.0 | 63.76 | 0.0 | 0.0 | 7.0 E |
| 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 |
| 34.0 | 102. | 32. | 40. | 22. | 0.0 | 397600. |
| 54.0 | 30. | 84. | 36. | 0.0 | .4 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 50.0 |
| .040 | 2000000000. | 20.0 | 4.0 | 60.0 | 52.0 | 12000. |
| .35 | 0.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|----------|-------|-----|-------|---------|-----|---------|
| 5607000. | .0825 | .18 | 18.0 | 372000. | 2.0 | 1.0 |
| 1540. | 0.0 | 0.0 | 63.76 | 0.0 | 0.0 | 7.0 M |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 34.0 | 102. | 32. | 40. | 22. | 0.0 | 397600. |
| 54.0 | 30. | 84. | 36. | 0.0 | .4 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 50.0 |

TABLE II
PROGRAM INPUTS FOR SATURN S-IC

| | | | | | | |
|----------|------------|------|---------|---------|------|---------|
| .040 | 200000000. | 20.0 | 4.0 | 60.0 | 52.0 | 12000. |
| .35 | 0.0 | 0.0 | 150000. | .25 | 100. | 400. |
| 5607000. | .0825 | .18 | 18.0 | 372000. | 2.0 | 1.0 |
| 1540. | 0.0 | 0.0 | 63.76 | 0.0 | 0.0 | 10. |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 34.0 | 102. | 32. | 40. | 22. | 0.0 | 397600. |
| 54.0 | 30. | 84. | 36. | 0.0 | .4 | |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 50.0 |
| .040 | 200000000. | 20.0 | 4.0 | 60.0 | 52.0 | 12000. |
| .35 | 0.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| 540000. | .168 | .2545 | 18.0 | 17650. | 2.0 | 1.0 |
| 3500. | 0.0 | 0.0 | 11.62 | 0.0 | 0.0 | 2.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 220600. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .045 | 200000000.20.0 | | 4.0 | 110.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|--------|---------|--------|------|---------|
| 540000. | .168 | .2545 | 18.0 | 17650. | 2.0 | 1.0 |
| 1500.0 | 500.0 | 4500.0 | 10.0 | 1.0 | 14.0 | 5.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 1.0 | 220600. |
| 42.0 | 24.0 | 72.0 | 36.0 | 1.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 1.0 | 1.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 37.0 |
| .045 | 200000000.20.0 | | 4.0 | 110.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| 540000. | .168 | .2545 | 18.0 | 17650. | 2.0 | 1.0 |
| 3500. | 0.0 | 0.0 | 11.62 | 0.0 | 0.0 | 8.0.E |
| 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 220600. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .045 | 200000000.20.0 | | 4.0 | 110.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|------|-------|-------|--------|-----|---------|
| 540000. | .168 | .2545 | 18.0 | 17650. | 2.0 | 1.0 |
| 3500. | 0.0 | 0.0 | 11.62 | 0.0 | 0.0 | 8.0.M |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 220600. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |

TABLE III
PROGRAM INPUTS FOR SATURN S-II

| | | | | | | |
|---------|----------------|------|---------|-------|------|--------|
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .045 | 200000000.20.0 | | 4.0 | 110.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| 540000. | .168 | .2545 | 18.0 | 17650. | 2.0 | 1.0 |
| 3500. | 0.0 | 0.0 | 11.62 | 0.0 | 0.0 | 11.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 220600. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .045 | 200000000.20.0 | | 4.0 | 110.0 | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| 500000. | .1395 | .2445 | 18.0 | 17400. | 2.0 | 1.0 |
| 3650. | 0.0 | 0.0 | 11.87 | 0.0 | 0.0 | 3.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 222700. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .100 | 200000000.20.0 | | 1.0 | 320. | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|--------|---------|--------|------|---------|
| 500000. | .1395 | .2445 | 18.0 | 17400. | 2.0 | 1.0 |
| 1500.0 | 500.0 | 4500.0 | 10.0 | 1.0 | 14.0 | 6.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 1.0 | 222700. |
| 42.0 | 24.0 | 72.0 | 36.0 | 1.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 1.0 | 1.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 37.0 |
| .100 | 200000000.20.0 | | 1.0 | 320. | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| 500000. | .1395 | .2445 | 18.0 | 17400. | 2.0 | 1.0 |
| 3650. | 0.0 | 0.0 | 11.87 | 0.0 | 0.0 | 9.0 E. |
| 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 222700. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .100 | 200000000.20.0 | | 1.0 | 320. | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

| | | | | | | |
|---------|-------|-------|-------|--------|-----|---------|
| 500000. | .1395 | .2445 | 18.0 | 17400. | 2.0 | 1.0 |
| 3650. | 0.0 | 0.0 | 11.87 | 0.0 | 0.0 | 9.0 M |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 222700. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |

TABLE IV
PROGRAM INPUTS FOR SATURN S-IVB

| | | | | | | |
|---------|----------------|-------|---------|--------|------|---------|
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .100 | 200000000.20.0 | | 1.0 | 320. | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |
| | | | | | | |
| 500000. | .1395 | .2445 | 18.0 | 17400. | 2.0 | 1.0 |
| 3650. | 0.0 | 0.0 | 11.87 | 0.0 | 0.0 | 12.0 |
| 1.0 | 1.0 | 0.0 | 1.0 | 1.0 | 0.0 | 1.0 |
| 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 6.31 | 27.8 | 5.81 | 18.14 | 4.5 | 0.0 | 222700. |
| 42.0 | 24.0 | 72.0 | 36.0 | 0.0 | .4 | |
| .1 | .262 | .262 | 243.0 | 117.0 | 1.0 | 1.0 |
| 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 1.0 | 0.2 | 0.0 | 0.0 | 1.0 |
| 0.01333 | 10.0 | 1.0 | 1.0 | 1.0 | 1.0 | .0314 |
| 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 37.0 |
| .100 | 200000000.20.0 | | 1.0 | 320. | 52.0 | 12000. |
| .35 | 250.0 | 0.0 | 150000. | .25 | 100. | 400. |

TABLE IV (CONTINUED)

D. Program Output and Results

The operator of the complete program will have an option of either receiving final results of the program only or receiving final results and intermediate results from each individual component. This option will be exercised by the operator by use of Sense Switch #1 on the computer. If Sense Switch #1 is turned on at the initial start of deck number one of the actuator-truss program, and left on during each succeeding program (tubing program, pump program, etc.) intermediate information will be composed of system pressure, moment arm, component weight, and component failure rate for each of the separate components within the system. This intermediate information will be useful to the operator if he wishes to examine a system by components as an aid and only an aid to the final overall program results. This will help him understand how an individual component is effected by a change in system pressure or moment arm. A cautionary note must be noted at this point that the intermediate results are inconclusive and false conclusions can be drawn from them. They in themselves are only indications of the components within the system and not indicative of the system as a whole.

The forms that output results are received, for both intermediate component information and final results, is explained below.

1. Final Results

All data received in the final results will be in the Fortran II-D floating-point constant form. This data will be printed out in three groups and will be in the following order.

The first group will be composed of program and system identification information to be used as reference after the program has been completed. This identification information will appear in the first two rows. The first row will have the words "Computer Program Number" at the start with the operator assigned input identification digit (VPRON) immediately following. The second row will be composed of two numbers, the first being the system pressure (psi), and the second being the moment arm (inches) for the system under investigation.

The second group of information to be printed out will be total system parameters. These parameters will be listed in three rows and are as follows:

Row #1

Row one will first contain the words "Total System Cost Is" and will be followed by the actual total system cost in dollars.

Row #2

Row two will first contain the words "Total System Weight Is----" and will be followed by the actual total weight of the system under consideration in pounds.

Row #3

Row three will first contain the words "Total System Generic Failure Rate Is" and will then be followed by the actual system failure rate given in failures per million hours.

This second group of information is designed to give the investigator an overall insight into the system as a complete hydraulic unit.

The third and last group of information to be printed out for the system under consideration, will be the individual component data of which the complete hydraulic system is composed. Information for the following components will be listed.

Row #1 - Actuator Data

Row one will contain the following data, and in the order listed, for a single actuator in the system under consideration 1) actuator weight (pounds), 2) actuator failure rate (failures per million hours) 3) actuator life (total "on" time - hours) 4) actuator life (total "on" time - cycles).

Row #2 - Fixed Angle Pump Data

The second row will contain the following fixed angle pump data, and in the order listed, for a single fixed angle pump. 1) fixed angle pump weight (pounds), 2) fixed angle pump failure rate (failures per million hours), 3) fixed angle pump life (total "on" time - hours).

Row #4 - Wobble Plate Pump or Intensifier

Row three will contain information concerning the wobble plate pump or hydraulic intensifier depending upon which one was used in the system under investigation. This information will be as follows and

in the order listed. 1) wobble plate pump or intensifier weight (pounds), 2) wobble plate pump or intensifier failure rate (failures per million hours), 3) wobble plate pump or intensifier life (total "on" time - hours).

Row #4 - Truss Data

The fourth row will contain the following truss data, and in the order as listed. 1) truss weight (pounds) 2) truss failure rate (failures per million hours).

Row #5 - Reservoir and/or Accumulator Data

The information being printed out in row number five will be 1) weight (pounds), and 2) failure rate (failures per million hours) for the reservoir, reservoir and accumulator, or accumulator depending upon which component or components are used in the system under consideration.

This then comprises the final print out for a single data point for the system under investigation.

2. Intermediate Component Results

The format by which the intermediate results will be obtained will be identical to the form explained previously under Final Results. The intermediate results will be printed out at the end of each component program in the form of a single row of information per initial data point under investigation. Each row of information will be composed of the following four pieces of information; (1) Hydraulic system pressure (Pounds per square inch), (2)

Actuator moment arm (inches), (3) weight of a single component within the system under investigation (pounds), (4) failure rate of a single component within the system under investigation (failure/ 10^6 hours).

Only if sense switch #1 is on will intermediate component results be received by the operator as well as the final results.

E. General Instructions for Running Program

The instructions for running the program will best be described by steps, and using this method they are listed below. The important thing to remember is that the sequence of individual decks is most important and only if the complete program is run in proper sequence will be correct results be obtained.

Step #1

Both the "O" Ring weight Subroutine, and the "O" Ring Failure rate subroutine must be loaded on the permanent disk storage before the main section of the program can be used. The first step will then be to load these two subroutines into permanent disk storage so that they are accessible to the main program. Either of the two subroutine decks may be loaded first.

Step #2

The next step will be to prepare the initial parameter data for the particular system under investigation. This preparation was described fully in Section C; Initial Data Input. The instructions there should be followed at this time. The final results of this step should be

thirteen (13) data cards in the proper sequence as explained in Section C.

Step #3

The number one (1) deck of the four (4) decks of the Actuator-Truss program should be run at this time using the data cards prepared in Step #2 as the input data to this deck. The end results of this deck will be only punched data cards.

Step #4

The number two (2) deck of the four (4) decks of the Actuator-Truss program will be run in this step using the punched data cards that were the results of deck one (1) from Step #3. As in Step #3 the end results of this deck will be punched data cards.

Step #5

The number three (3) deck of the four (4) decks of the Actuator Truss program will be run in this step using the punched data cards that were the results of deck two (2) from Step #4. As in Step #3 and Step #4 the end results of this deck will be punched data cards.

Step #6

The number four (4) deck of the four (4) decks of the Actuator-Truss program will be run in this step using the punched data cards that were the results of deck three (3) from Step #5. The results of this run will be punched data cards and intermediate results of the actuator and truss if Sense Switch #1 has been turned

on. If Sense Switch #1 has not been turned on, the results of this step will only be punched data cards. If intermediate data has been printed it will be (1) system pressure (psi), (2) moment arm (in.), (3) actuator weight (lbs.), (4) actuator generic failure rate (failures/ 10^6 hours), (5) truss weight (lbs.), (6) truss generic failure rate (failures/ 10^6 hours). This intermediate information will be in the form described in Section D; Program Output and Results.

Step #7

In this step, the tubing deck will be run using the punched data cards from Step #6 as input data. The results of this step will be punched data cards and intermediate tubing results if Sense Switch #1 has been turned on. If Sense Switch #1 has not been turned on, the results of this step will only be punched data cards. If intermediate tubing data is printed, it will be, (1) system pressure (psi), (2) moment arm (in.), (3) combined tubing and fitting weight (lbs.), (4) tubing system generic failure rate (failures/ 10^6 hours). As in Step #6 the intermediate information will be in the form described in Section D.

Step #8

The number one (1) deck of the two (2) decks of the fixed angle pump program will be run in this step using the punched data cards that were the results of the tubing program in Step #7 as input data. The end results

of this step will only be punched data cards.

Step #9

The number two (2) deck of the two (2) decks of the fixed angle pump program will be run in this step using the punched data cards produced in Step #8 as input data. The results of this step will be punched data cards, and intermediate fixed angle pump information but only if Sense Switch #1 has been turned on. If Sense Switch #1 has not been turned on, the only results of this step will be punched data cards. If the option has been exercised to receive intermediate information on the fixed angle pump, it will be printed in a single row per system data point and will be (1) System pressure (psi), (2) moment arm (in.), (3) a single fixed angle pump weight (lbs.), (4) a single fixed angle pump generic failure rate (failures/ 10^6 hours). This pump information will be of the form described in Section D.

Step #10

In this step the wobble plate pump program will be run using the punched output data cards from Step #9 as input data. The results of this step will be punched data cards and intermediate wobble plate pump information, if Sense Switch #1 has been turned on. If Sense Switch #1 has not been turned on, the only results of this step will be punched data cards. The intermediate wobble plate information will be a single row of data

per system data point. This row of data will be in the order of (1) system pressure (psi), (2) Moment arm (in.), (3) weight of a single wobble plate pump (lbs.) (4) generic failure rate of a single wobble plate pump (failures/ 10^6 hours). As with all intermediate component information it will be in the form described in Section D, Program Output and Results.

Step #11

The hydraulic intensifier program will be run in this step using the punched output data cards from the wobble plate pump program, (Step #10), as input data. If Sense Switch #1 has been turned on, the results of the intensifier program will be both punched output data cards, and intermediate hydraulic intensifier information. If Sense Switch #1 is not turned on, the results of the run will only be in the form of punched output data cards. Intermediate intensifier information will be in the form of a single row of intensifier data for each system data point. Each row will be arranged in the following sequence: (1) system pressure (psi), (2) moment arm (in.), (3) weight of a single intensifier (lbs.), (4) generic failure rate of a single intensifier (failures/ 10^6 hours). The intermediate information will be in the form described in Section D.

Step #12

The filter program is run in this step using the punched output data cards from the hydraulic intensifier program (Step #11) as input data. If Sense Switch #1 has been turned on, the results of the filter program will be both punched output data cards, and intermediate filter information. If Sense Switch #1 is not turned on, the results of the filter program will only be in the form punched output data cards. Intermediate information will be in the form of a single row of filter data for each system data point. Each row will be arranged in the following sequence; (1) system pressure (psi), (2) moment arm (in.), (3) weight of a single filter for the system being investigation (lbs.), (4) generic failure rate of a single filter (failures/ 10^6 hours). The intermediate filter information will be in the form described in Section D.

Step #13

The reservoir-accumulator program deck will be run in this step. The input data will be the punched output data cards from the filter program (Step #12). The results of the reservoir-accumulator program will be both punched output data cards and intermediate reservoir-accumulator information, if Sense Switch #1 has been turned on. If Sense Switch #1 has not been turned on, the results of the reservoir-accumulator program will

only be in the form of punched output data cards. Reservoir-accumulator intermediate information will be in the form of a single row of data for every system data point. Each row of information will be in the following form; (1) system pressure (psi), (2) moment arm (in.), (3) weight of a single reservoir and/or accumulator (lbs.), (4) generic failure rate of a single reservoir and/or accumulator (Failures/ 10^6 hours). The output form described in Section D, will be used for the reservoir-accumulator information.

Step #14

The final program, complete vehicle conversion deck, will finalize the running of the program. The punched output data cards from the reservoir-accumulator program (Step #13) will be used as input data. The final results will be printed out of the end of the program and will be comprised of information described in Section D; Program Output and Results, Part 1: Final Results.

This completes the running of the program and will give the user a complete set of information for varying moment arms and varying system pressures for a particular hydraulic system under investigation.

F. Listing of Component Computer Programs

Following is a complete listing of the individual component and subroutine programs which form the overall final study program.

They are listed in the order to be used when an investigation for a particular hydraulic system is conducted.

```

*LDISK
      SUBROUTINE OSWE (DIAM,J,RINGWT)
      IF (J-1)101,101,102
101  IF (DIAM-4.475)103,104,104
104  WIDH=2.75E-1
      GO TO 70
103  IF (DIAM-1.475)105,106,106
106  WIDH=2.1E-1
      GO TO 70
105  IF (DIAM-.734)107,108,108
108  WIDH=1.39E-1
      GO TO 70
107  IF (DIAM-.362)109,110,110
110  WIDH=1.03E-1
      GO TO 70
109  IF (DIAM-.070)111,112,112
112  WIDH=7.0E-2
      GO TO 70
111  IF (DIAM-.050)113,114,114
114  WIDH=6.0E-2
      GO TO 70
113  IF (DIAM-.036)115,116,116
116  WIDH=5.0E-2
      GO TO 70
115  WIDH=4.0E-2
      GO TO 70
102  IF (DIAM-4.96)121,122,122
122  WIDH=2.75E-1
      GO TO 71
121  IF (DIAM-1.828)123,124,124
124  WIDH=2.1E-1
      GO TO 71
123  IF (DIAM-.978)125,126,126
126  WIDH=1.39E-1
      GO TO 71
125  IF (DIAM-.536)127,128,128
128  WIDH=1.03E-1
      GO TO 71
127  IF (DIAM-.210)129,130,130
130  WIDH=7.0E-2
      GO TO 71
129  IF (DIAM-.159)131,132,132
132  WIDH=6.0E-2
      GO TO 71
131  IF (DIAM-.126)133,134,134
134  WIDH=5.0E-2
      GO TO 71
133  WIDH=4.0E-2
      GO TO 71
70  DIAI=DIAM
      DIAO=DIAM+(2.0*WIDH)
      GO TO 75
71  DIAO=DIAM
      DIAI=DIAM-(2.0*WIDH)
75  RINGWT=.0574*(DIAI+DIAO)*(WIDH**2.0)
      RETURN
      END

```

TABLE V
O-RING WEIGHT SUBROUTINE
COMPUTER PROGRAM LISTING

```

*LDISK
      SUBROUTINE OSFR (DIAM,SPRE,I,J,RINGFR)
      IF (J-1)1,1,2
1 IF (DIAM-4.475)3,4,4
4 ALPH=1.28E-1
  CLER=5.0E-3
  WIDH=2.75E-1
  GO TO 50
3 IF (DIAM-1.475)5,6,6
6 ALPH=1.09E-1
  CLER=3.5E-3
  WIDH=2.1E-1
  GO TO 50
5 IF (DIAM-.734)7,8,8
8 ALPH=1.18E-1
  CLER=3.0E-3
  WIDH=1.39E-1
  GO TO 50
7 IF (DIAM-.362)9,10,10
10 ALPH=1.3E-1
  CLER=2.5E-3
  WIDH=1.03E-1
  GO TO 50
9 IF (DIAM-.070)11,12,12
12 ALPH=1.88E-1
  CLER=2.0E-3
  WIDH=7.0E-2
  GO TO 50
11 IF (DIAM-.050)13,14,14
14 ALPH=1.9E-1
  CLER=2.0E-3
  WIDH=6.0E-2
  GO TO 50
13 IF (DIAM-.036)15,16,16
16 ALPH=1.9E-1
  CLER=2.0E-3
  WIDH=5.0E-2
  GO TO 50
15 ALPH=2.08E-1
  CLER=2.0E-3
  WIDH=4.0E-2
  GO TO 50
2 IF (DIAM-4.96)21,22,22
22 ALPH=1.28E-1
  CLER=5.0E-3
  WIDH=2.75E-1
  GO TO 51
21 IF (DIAM-1.828)23,24,24
24 ALPH=1.09E-1
  CLER=3.5E-3
  WIDH=2.1E-1
  GO TO 51
23 IF (DIAM-.978)25,26,26
26 ALPH=1.18E-1
  CLER=3.0E-3
  WIDH=1.39E-1
  GO TO 51

```

TABLE VI
O-RING FAILURE RATE SUBROUTINE
COMPUTER PROGRAM LISTING

```

25 IF (DIAM-.536)27,26,28
26 ALPH=1.3E-1
27 CLER=2.5E-3
28 WIDTH=1.03E-1
29 GO TO 51
30 IF (DIAM-.210)29,30,30
31 ALPH=1.88E-1
32 CLER=2.0E-3
33 WIDTH=7.0E-2
34 GO TO 51
35 IF (DIAM-.159)31,32,32
36 ALPH=1.9E-1
37 CLER=2.0E-3
38 WIDTH=6.0E-2
39 GO TO 51
40 IF (DIAM-.126)33,34,34
41 ALPH=1.9E-1
42 CLER=2.0E-3
43 WIDTH=5.0E-2
44 GO TO 51
45 ALPH=2.08E-1
46 CLER=2.0E-3
47 WIDTH=4.0E-2
48 GO TO 51
49 DIAI=DIAM
50 DIAO=DIAM+(2.0*WIDTH)
51 GO TO 52
52 DIAO=DIAM-(2.0*WIDTH)
53 DO 53 K=1,7,1
54 IF (I-K)54,54,53
55 CONTINUE
56 GO TO (61,62,63,64,65,66,67),K
57 ORINGFR=((((DIAI+DIAO)*(ALPH*WIDTH)*(1.0-ALPH/2.0))+  

58 1(1.512E-1/(((DIAI+DIAO)**2.0)*ALPH))+  

59 2(2.3E-3/CLER)+  

60 3(2.91E+1*ALPH*WIDTH))*  

61 4(((1.835E-1*SPRE)/(4.0E+4-SPRE))+(7.04E+1/(2.0E+3+SPRE)))  

62 RETURN
63 ORINGFR=((((DIAI+DIAO)*(ALPH*WIDTH)*(1.0-ALPH/2.0))+  

64 1(1.512E-1/(((DIAI+DIAO)**2.0)*ALPH))+  

65 2(2.3E-3/CLER)+  

66 3(((1.529E-1*SPRE)/(4.0E+4-SPRE))+(4.69E+1/(2.0E+3+SPRE)))  

67 RETURN
68 ORINGFR=((((DIAI+DIAO)*(ALPH*WIDTH)*(1.0-ALPH/2.0))+  

69 1(1.512E-1/(((DIAI+DIAO)**2.0)*ALPH))+  

70 2(((1.529E-1*SPRE)/(4.0E+4-SPRE))+(4.69E+1/(2.0E+3+SPRE)))  

71 RETURN
72 END

```

TABLE VI (CONTINUED)

```

803 READ 820,TORQ,VELS,TRAAC,AKVEL,EINT,ANUMB,AVPR
      READ 820,PREI,DPRE,PREM,AMOM,DMOM,AMAX,VPRON
      READ 820,AAAA1,AAA2,AAA3,AAA4,AAA5,AAA6,AAA7
      READ 820,AAA8,AAA9,AAA10,AIPA1,AIPA2,AIPA3,AIPA4
      READ 820,XMDC,XMD,XMDA,XMDE,XXXX1,AKENG
      READ 820,S1,S2,S3,S4,XXXX2,PPPP9
      READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
      READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
      READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
      READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
      READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
      READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
      READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
      X=1.0
      APPPI=1.0
      VELA=AVPR*VELS
800 PRES=PREI
801 WEIGHT=0.0
      RELIAB=0.0
      VELL=VELA*AMOM
      TRAL=TRAAC*AMOM
      FORC=TORQ/AMOM
      FLOW=FORC*VELL/PRES
      VELX=VELS*AMOM
      FLOA=FORC*VELX/PRES
      AV2F1=((VELA*TORQ)**0.5)/(PRES**0.75)
      AV2F2=1.0+.000308*PRES
      AV2FV=.0594*AV2F1
      AV2FJ=.309*AV2F1
      AV2FB=.0242*AV2F1*PRES**0.5
      AV2FI=.308*AV2F2*AV2F1+.118*(AV2F2*AV2F1)**0.5
      AV2FX=1.562*AV2F1+.575*(AV2F2*AV2F1)**0.5
      AV2FW=.217*(AV2FI**2.0-AV2FJ**2.0)*AV2FX
      WEIGHT=WEIGHT+.217*(AV2FI**2.0-AV2FJ**2.0)*AV2FX
      RELIAB=RELIAB+.00227*(PRES**0.5/AV2F1)+2.27/(AV2F1*PRES**0.5)
      WEIGHT=WEIGHT+.0215*AV2FJ*AV2FX
      RELIAB=RELIAB+.00227*(PRES**0.5/AV2F1)+2.27/(AV2F1*PRES**0.5)
      AV12J=.912*AV2FJ**1.5/PRES**.25
      AV12X=25.7*AV12J
      AV12I=6.14*AV12J
      AV12W=215.*AV12J**3.0
      AV12R=.335*(PRES**0.5)*(AV12J**2.0)*(1.0/(AV12J**1.2)-6.9)
      IF (AV12R-.0001)815,815,816
815 AV12R=.0001
816 WEIGHT=WEIGHT+.088*PRES*AV12J**3.0
      RELIAB=RELIAB+.00041*AV12J*PRES
      AV1MC=AV12J**3.0*PRES
      AV1MI=4.19*AV12J
      AV1FW=.00143*AV1MC/AV12J
      WEIGHT=WEIGHT+.516*AV1MC+3.86*AV1FW+.00437*PRES*AV1MI**3.0
      RELIAB=RELIAB+.00026/AV1MC+36.5/(PRES*AV1MI)+7.53*AV12J**2.0+.02
      AV1DR=.00248/AV1FW**.333
      RELIAB=RELIAB+.00286/AV1FW**.333+.0008+.0182*AV1DR+.364*AV1DR
      WEIGHT=WEIGHT+1.104*AV1FW+.00306+.23*AV1FW+.572*AV1FW
      AV1NW=.43*AV12W
      WEIGHT=WEIGHT+3.90*AV1MI**3.0+.43*AV12W+2.2*AV1NW+.00152
      RELIAB=RELIAB+.35*AV1MI+.337*AV12R+.000258/AV1NW**0.333

```

TABLE VII
DECK NUMBER 1 - ACTUATOR
COMPUTER PROGRAM LISTING

```

AV1NI=.692*AV12I
CALL OSWE (1.925*AV1MI,1,RINGWT)
WEIGHT=WEIGHT+RINGWT
CALL OSWE (4.0*AV1NI,2,RINGWT)
WEIGHT=WEIGHT+RINGWT
CALL OSFR (.176,PRES,5,2,RINGFR)
RELIAB=RELIAB+2.86*RINGFR
CALL OSFR (1.925*AV1MI,0.1*PRES,6,1,RINGFR)
RELIAB=RELIAB+RINGFR
CALL OSFR (4.0*AV1NI,0.1*PRES,5,2,RINGFR)
RELIAB=RELIAB+RINGFR
AV1IW=.00402*AV2FJ**2.0
AV1IR=.042*AV2FJ**2.0/AV12J
AV1II=7.93*AV2FJ**2.0/AV2FX
AV1YW=.104*AV1II**3.0
AV1YR=1.01*AV12R
AV1YI=1.01*AV1II
CALL OSWE (1.01*AV1YI,2,RINGWT)
AV1XW=2.0*RINGWT
CALL OSWE (1.275*AV1YI,2,RINGWT)
AV1XW=AV1XW+RINGWT
CALL OSFR (1.01*AV1YI,PRES/2.0,5,2,RINGFR)
AV1XR=RINGFR
CALL OSFR (1.01*AV1YI,PRES/2.0,2,2,RINGFR)
AV1XR=AV1XR+RINGFR
CALL OSFR (1.275*AV1YI,PRES/2.0,2,2,RINGFR)
AV1XR=AV1XR+RINGFR
AV1WW=.000215*PRES*AV1II**3.0
AV1WR=.000322/AV1II
AV2CW=2.12E-7*(AV2FJ**3.0)*(PRES**1.5)
AV2CR=.000025/AV2FV
WEIGHT=WEIGHT+.446*AV2CW
CALL OSWE (1.015*AV2FI,2,RINGWT)
WEIGHT=WEIGHT+6.0*RINGWT
CALL OSFR (1.015*AV2FI,PRES/2.0,2,2,RINGFR)
RELIAB=RELIAB+2.4*AV2CR+6.0*RINGFR
AV2NW=.000356*AV2FV*PRES*AV2FJ**2.0*AAAA1
AV2NR=AAAA1*(5.22E-6)*PRES**0.5/(AV2FJ*AV2FV)
AV2NI=AAAA1*AV2FJ*(.615+.000128*PRES)
CALL OSWE (1.39*AV2FJ,1,RINGWT)
WEIGHT=WEIGHT+.352*AV2NW+2.0*AAAA1*RINGWT
CALL OSFR (1.39*AV2FJ,0.0,6,1,RINGFR)
RELIAB=RELIAB+AAAA1*(.00052/AV2FJ+2.0*RINGFR)
WEIGHT=WEIGHT+2.5E-9*(AV2FJ**4.0)*(PRES**2.0)/AV2FV*2.0
RELIAB=RELIAB+1.71/((AV2FJ)*(PRES**0.5))*2.0
AV2BX=.0000444*PRES*AV2FJ**2.0/AV2FV
CALL OSWE (1.12*AV2NI,2,RINGWT)
CALL OSFR (1.12*AV2NI,PRES/2.0,5,2,RINGFR)
WEIGHT=WEIGHT+4.0*AAAA1*RINGWT
RELIAB=RELIAB+4.0*AAAA1*RINGFR
CALL OSWE (1.015*AV2FI,2,RINGWT)
CALL OSFR (1.015*AV2FI,PRES/2.0,5,2,RINGFR)
WEIGHT=WEIGHT+2.0*RINGWT
RELIAB=RELIAB+2.0*RINGFR
AV2K1=1.18*((AV2BX)+2.0*AV2FV)+3.0*AAAA1*AV2FV
AV2KX=1.31*AV2K1+.000128*AV2FJ*PRES
AV2KR=.0468/(AV2FV*AV2FJ*PRES)
AV2KW=.000223*AV2K1*PRES*AV2FJ**2.0
WEIGHT=WEIGHT+.1685*AV2KW

```

```

RELIAB=RELIAB+.1165*AV2KR+.00001/AV2FV
AV2HW=.0434*AV2FJ**3.0
WEIGHT=WEIGHT+.765*AV2HW
RELIAB=RELIAB+.000915/(AV2HW**.333)
CALL OSWE (.92*AV2FJ,2,RINGWT)
CALL OSFR (.92*AV2FJ,PRES/2.0,2,2,RINGFR)
WEIGHT=WEIGHT+RINGWT+4.37*AAAA2*(AV12J**3.0)*(PRES**.333)
RELIAB=RELIAB+RINGFR
AVMJR=.0517/((AV12J)*PRES**.333)*AAAA2
AVMJK=22.65*AV12J*AAAA2
WEIGHT=WEIGHT+AAAA2*24.6*AV12J**3.0*PRES**.333
AVMKR=.0982*AVMJR
AVMAW=AAAA2*26.05*AV2FX*AV12J**2.0
AVMAR=AAAA2*AV2FX*.001193*AV2FX/AV12J
WEIGHT=WEIGHT+.837*AVMAW
RELIAB=RELIAB+.00202*AAAA2*(1.0/(AVMAW**.333+1.0E-20))
CALL OSWE(1.31*AV2FJ,1,RINGWT)
CALL OSFR(1.31*AV2FJ,40.,6,1,RINGFR)
WEIGHT=WEIGHT+RINGWT*4.0
RELIAB=RELIAB+RINGFR
CALL OSFR(1.31*AV2FJ,PRES,6,1,RINGFR)
RELIAB=RELIAB+RINGFR
CALL OSFR(1.31*AV2FJ,PRES/2.0,6,1,RINGFR)
RELIAB=RELIAB+RINGFR*2.0
WEIGHT=WEIGHT+2.0*AV12W+4.0*AV2CW+2.0*AV2NW+2.0*AV2KW+AV2HW
ORELIAB=RELIAB+2.0*AV12R+2.0*AV1DR+4.0*AV2CR+2.0*AV2NR+2.0*AV2KR+
12.0*AVMJR+AVMKR
PUNCH 830,AV2FW,AV2F2,AV2FV,AV2FJ,AV2FB,AV2FI,AV2FX
PUNCH 830,AV12J,AV12X,AV12I,AV1MI,AV1FW,AV11W,AV11R
PUNCH 830,AV1YW,AV1YR,AV1YI,AV1XW,AV1XR,AV1WW,AV1WR
PUNCH 830,AVMAW,AVMAR,TORQ,AAAA8,TRA, EINT,WEIGHT
PUNCH 830,RELIAB,AAAA1,AAAA2,AAAA3,AAAA4,AAAA5,VELS
PUNCH 830,AAAA6,AIPA1,AIPA2,AIPA3,AIPA4,APPPI,VELX
PUNCH 830,AMOM,PRÉS,AAAA7,TRAL,FORC,FLOW,FLOA
PUNCH 830,AKVEL,ANUMB,XMDC,XMDD,XMDB,XMDA,XMDE
PUNCH 830,AAA9,AAA10,XXXX1,AKENG,VPRON
PUNCH 830,S1,S2,S3,S4,XXXX2,PPPP9
PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PRES=PRES+DPRE
IF (PREM-PRES)802,802,801
802 AMOM=AMOM+DMOM
IF (AMAX-AMOM)900,900,800
900 IF (X-4.0)902,901,901
902 X=X+2.0E-10
GO TO 803
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
901 STOP
END

```

TABLE VII (CONTINUED)

```

800 READ 820,AV2FW,AV2F2,AV2FV,AV2FJ,AV2FB,AV2FI,AV2FX
      READ 820,AV12J,AV12X,AV12I,AV1MI,AV1FW,AV11W,AV11R
      READ 820,AV1YW,AV1YR,AV1YI,AV1XW,AV1XR,AV1WW,AV1WR
      READ 820,AVMAW,AVMAR,TORO,AAAA8,TRAA,EINT,WEIGHT
      READ 820,RELIAB,AAAA1,AAAA2,AAAA3,AAAA4,AAAA5,VELA
      READ 820,AAAA6,AIPA1,AIPA2,AIPA3,AIPA4,APPPI,VELL
      READ 820,AMOM,PRES,AAAAT,TRAL,FORC,FLOA,FLOW
      READ 820,AKVEL,ANUMB,XMDC,XMDD,XMDB,XMDA,XMDE
      READ 820,AKENG,VPRON
      READ 820,S1,S2,S3,S4,XXXX1,XXXX2,PPPP9
      READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
      READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
      READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
      READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
      READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
      READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
      READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
      APRBW=.361E-4*FORC-.0167
      APRBJ=.0074*FORC**0.5
      APRBY=APRBJ
      APRBI=AAA3*(1.25*APRBJ+.125)
      APRBR=.0343/APRBJ
      APRRI=(1.5*APRBJ+.25)*(1.0-AAA3)
      RELIAB=RELIAB+.0256*APRRI+.0041*(2.5*APRBJ+.25)+.0437/APRBJ
      APRHK=APRBJ
      OWEIGHT=WEIGHT+.0907*APRBJ**2.0*(2.94*(APRBI+APRRI)+APRBJ)+.761*
      1APRBJ**3.0+APRBW*(1.0-AAA3)+.0903*APRHK**3.00
      8100APPPK=(1./9.464)*(APRHK+AAA8*.00159*VELL**2./APRHK+TRAL+.71
      1*APPPI**.5)
      IF (ABSF(APPPK-APPPI)-.00001*APPPK)811,812,812
      812 APPPI=(APPPK+APPPI)/2.0
      GO TO 810
      811 IF (APPPK**2.0-.675*APRHK)813,813,817
      817 APPPJ=(APPPK**2.0-.675*APRHK)**0.5
      IF (APPJ-0.812)813,814,814
      813 APPPJ=0.812
      APPPK=(APPJ**2.0+.675*APRHK)**0.5
      814 APPPI=(1.273*FORC/PRES+APPPK**2.0)**0.5
      APPPY=.305*APPPI**0.5
      OAPPPX=APRHK+1.42*APPPK**.5+.536*APPPK+.00318*VELL**2./APRHK*
      1AAA8+2.0*TRAL+.305*APPPI**0.5
      WEIGHT=WEIGHT+.176*APPPY*(APPPI**2.0-APPPK**2.0)+.146*APRHK*APPPX
      CALL OSFR (.97*APPPI,0.1*PRES,2,2,RINGFR)
      CALL OSWE (.97*APPPI,2,RINGWT)
      WEIGHT=WEIGHT+RINGWT+.887*RINGWT
      RELIAB=RELIAB+.258*APPPK/APRHK+.00175/APRHK+.0001+RINGFR
      CALL OSFR (APPPI,0.1*PRES,7,2,RINGFR)
      RELIAB=RELIAB+RINGFR+.809*((PRES/(VELL*FORC))**0.5)
      OWEIGHT=WEIGHT+1.505E-4*(VELL**2.0/APRHK)*(APPPK*(VELL*FORC/PRES)
      1**.5+.0289*VELL*FORC/PRES)
      ACBBI=APPPI+(4.65E-5)*APPPI*PRES
      ACBBY=ACBBI+(13.06E-5)*APPPI*PRES
      OWEIGHT=WEIGHT+.225*((TRAL+.0032*AAA8*VELL**2./APRHK+APPPY)*(ACBBI
      1**2.0-APPPI**2.0)+(.71*APPPK**.5+.536*APPPK)*(31.08E-5*APPPK**2.0
      2*PRES+241.5E-10*(APPPK*PRES)**2.0)+(7.77E-5*APPPK*PRES)*(ACBBI**
      32.0-(APPPK+15.54E-5*APPPK*PRES)**2.0)+4.98E-5*APPPI*PRES*(ACBBY**
      42.0-ACBBI**2.0))+.00293

```

TABLE VIII
DECK NUMBER 2 - ACTUATOR
COMPUTER PROGRAM LISTING

```

CALL OSWE (.239,2,RINGWT)
CALL OSFR (.239,PRES,2,2,RINGFR)
WEIGHT=WEIGHT+2.0*RINGWT
ACUUI=.774*AV2FI
WEIGHT=WEIGHT+.11*AV2FW
ORELIAB=RELIAB+(.019*(TRAL+(.00318*AAAA8*(VELL**2.0)/APRHK+APPPY)
1*APPI*(PRES**1.5))/(FORC*VELL))+.001+2.0*RINGFR+.00139/
2(ACUUI**0.5)
CALL OSWE (ACUUI,2,RINGWT)
CALL OSFR (ACUUI,PRES,2,2,RINGFR)
RELIAB=RELIAB+RINGFR*2.0
WEIGHT=WEIGHT+2.0*RINGWT
ACBSW=(2.66E-14)*(APPI*PRES)**3.0
ACBSR=16.08/(APPI*PRES)
CALL OSWE (ACBBI,2,RINGWT)
CALL OSFR (ACBBI,PRES,2,2,RINGFR)
WEIGHT=WEIGHT+RINGWT
RELIAB=RELIAB+RINGFR
CALL OSWE (.94*APPK,1,RINGWT)
CALL OSFR (.94*APPK,PRES/2.0,5,2,RINGFR)
ACSOW=RINGWT
ACSOR=RINGFR
ACSCW=.758*ACSOW
CALL OSFR (APPK,PRES/2.0,3,1,RINGFR)
ACSCR=RINGFR
WEIGHT=WEIGHT+3.39*ACSCW+.387+.0065
RELIAB=RELIAB+.0339*APPK
CALL OSFR (.239,PRES,6,1,RINGFR)
RELIAB=RELIAB+.730+2.0*RINGFR
CALL OSFR (.145,PRES,5,1,RINGFR)
RELIAB=RELIAB+RINGFR
AVDBW=AAAA4*(AV11W+3.0*AV1YW+AV1WW+AV1XW)
AVDBR=AAAA4*(AV11R+AV1YR+AV1XR+AV1WR)
WEIGHT=WEIGHT+2.0*AV1FW+AV11W+2.0*AV1YW+AV1WW+AV1XW+AVMAW+APRBW
RELIAB=RELIAB+AV11R+2.0*AV1YR+AV1XR+AV1WR+AVMAR+AVDBR+APRB
AFCCI=0.0
ABFHJ=0.0
PUNCH 830,VELA,TORQ,PRES,AAAA7,AAAA8,WEIGHT,RELIAB
PUNCH 830,ACSOW,ACSCW,ACSOR,ACSCR,AMOM,APPJ,APPPY
PUNCH 830,APPI,FORC,AVDBW,AVDBR,ACBBI,APPK,AV2FJ
PUNCH 830,AV2FI,AV12I,AV12J,AV2FX,AV1YI,AV12X,TRAL
PUNCH 830,VELL,APRHK,APRBW,AVMAW,APRBR,APRBJ,AVMAR,ACBSR
PUNCH 830,ACBSW,ACBBY,APRRI,APRBI,APPPX,AV2FB,AV1XW
PUNCH 830,AAAA1,AAAA6,AAAA4,AIPA4
PUNCH 830,AKVEL,ANUMB,XMDC,XMDD,XMDA,XMDE
PUNCH 830,AAA2,AAA3,AAA5,AAA7,AAA8,AIPA1,AIPA2
PUNCH 830,AIPA3,FLOW,EINT,FLOA,AV2FJ,APRBJ,TORQ
PUNCH 830,AAA9,AAA10,ABFHJ,AFCCI,AKENG,XXXX1,VPRON
PUNCH 830,S1,S2,S3,S4,XXXX2,PPP9
PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
GO TO 800
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)

```

TABLE VIII (CONTINUED)

```

800 READ 820,VELA,TORQ,PRES,AAAA7,AAAA8,WEIGHT,RELIAB
READ 820,ACSW,ACSCW,ACSOR,ACSCR,AMOM,APPPJ,APPY
READ 820,APPPJ,FORC,AVDBW,AVDBR,ACBBI,APPPK,AV2FJ
READ 820,AV2FI,AV12I,AV12J,AV2FX,AV1YI,AV12X,TRAL
READ 820,VELL,APRHK,APRBW,AVMAW,APRBR,APRBJ,AVMAR,ACBSR
READ 820,ACBSW,ACBBY,APRRI,APRBI,APPPX,AV2FB,AV1XW
READ 820,AAAA1,AAAA6,AAAA4,AIPA4
READ 820,AKVEL,ANUNB,XMDC,XMDD,XMDB,XMDA,XMDE
READ 820,AAAA2,AAAA3,AAAA5,AAAA7,AAAA8,AIPA1,AIPA2
READ 820,AIPA3,FLOW,EINT,FLOA,AV2FJ,APRBJ,TORQ
READ 820,AAAA9,AAA10,ABFHJ,AFCCI,AKENG,XXXX1,VPRON
READ 820,S1,S2,S3,S4,XXXX2,PPPP9
READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
ABFAI=(.534+.9218E-4*PRES)*(VELA*TORQ/(PRES**1.5))**0.5
ABFAX=1.39*(VELA*TORQ)**0.5/(PRES**.75)
ABFAW=5.7E-5*(VELA*TORQ)**1.5/(PRES**1.25)
CALL OSFR (ABFAI,40.0,2,2,RINGFR)
RELIAB=RELIAB+(9.04*ABFAI*PRES/(VELA*TORQ)+RINGFR)*AAAA7
CALL OSFR (ABFAI,PRES,2,2,RINGFR)
RELIAB=RELIAB+RINGFR*AAAA7
CALL OSFR (.346*ABFAI,40.0,2,2,RINGFR)
RELIAB=RELIAB+RINGFR*AAAA7+.0039*AAAA8
ABPB1=.662*ABFAI
ABPBX=.898*ABFAX
ABPBW=.2365*ABFAW
CALL OSFR (ABPB1,PRES,2,2,RINGFR)
ABPB1=3.54*ABPB1*PRES/(VELA*TORQ)+RINGFR
CALL OSFR (.53*ABPB1,PRES,2,2,RINGFR)
ABPB1=ABPB1+RINGFR
CALL OSFR (ABPB1,1.5*PRES,5,2,RINGFR)
ABPB1=ABPB1+RINGFR
ABB1=.89*(VELA*TORQ)**0.5/PRES**.75
ABB1X=1.67*ABB1
WEIGHT=WEIGHT+.222*(ABB1**3.0)
CALL OSWE (ABB1,2,RINGWT)
CALL OSFR (ABB1,PRES,2,2,RINGFR)
WEIGHT=WEIGHT+RINGWT
RELIAB=RELIAB+RINGFR+.0375/ABB1
CALL OSWE (1.13*ABB1,2,RINGWT)
CALL OSFR (1.13*ABB1,40.0,2,2,RINGFR)
WEIGHT=WEIGHT+RINGWT+.0000697*PRES*ABB1**3.0+.0887+ACSW+ACSCW
RELIAB=RELIAB+RINGFR+30.6/(ABB1*PRES)+.0202+ACSOR+ACSCR
ABFHJ=(5.06E-7)*((VELA*TORQ)**1.5)/(PRES**0.5)
ABFHJ=.2185*(VELA*TORQ/PRES)**0.5
ABFHJ=ABFHJ+(7.5E-6)*(PRES*VELA*TORQ)**0.5
ABFEW=.0075*VELA*TORQ/PRES
WEIGHT=WEIGHT+.0028*ABFHJ**3.0
CALL OSWE (.625*ABFHJ,2,RINGWT)
CALL OSFR (.625*ABFHJ,20.0,2,2,RINGFR)
RELIAB=RELIAB+RINGFR
WEIGHT=WEIGHT+RINGWT

```

TABLE IX
DECK NUMBER 3 - ACTUATOR
COMPUTER PROGRAM LISTING

```

CALL OSWE (.97*ABFHI,1,RINGWT)
CALL OSFR (.97*ABFHI,PRES,5,1,RINGFR)
RELIAB=RELIAB+RINGFR
WEIGHT=WEIGHT+RINGWT
AVDAW=AAAA4*2.8*ABPBW
AVDAR=AAAA4*ABPBR
AVDAI=AAAA4*ABPB1
AVDAX=AAAA4*ABPBX
ABFAI=ABFAI*AAA7
ABFAX=ABFAX*AAA7
ABFAW=ABFAW*AAA7
ABPB1=ABPB1*AAA8
ABPBX=ABPBX*AAA8
ABPBW=ABPBW*AAA8
ABPBR=ABPBR*AAA8
ABHBX=1.3*ABFHI+1.70*ABBDI*(1.0+2.05E-4*PRES)
OABHBW=0.00000+.0864*ABFHI**3.*{(1.3*((2.0E-4)*PRES+(1.0E-8)*PRES
1**2.0)+.2195)+ABBDI**3.0*.4770+(4.44E-4)*PRES+(4.47E-8)*PRES**2
22.0+.0635+ABFAX*ABFAI**2.0*(.02365+(.561E-4)*PRES+(.718E-8)*PRES
3**2.0)+.0864*(ACBBI**2.0+((1.608E-4)*PRES+(.647E-8)*PRES**2.0)*
4ABHBX+(ACBBI**2.0-APPPK**2.0)*(ABHBX-1.3*ABFHI))
AVBPI=1.115*AV2FJ
CALL OSWE(AVBPI,2,RINGWT)
CALL OSFR(AVBPI,PRES,2,2,RINGFR)
WEIGHT=WEIGHT+.0324*AV2FJ**2.5+RINGWT+.00727*APPPK
RELIAB=RELIAB+.00196*AV2FJ**0.5+RINGFR+.015
AVBB2=AAA2
OAVBBX=AVBB2*(AV2FI*(1.0+(3.01E-4)*PRES)+AV12I*(1.0+(2.665E-4)*
1PRES)+58.3*AV12J)
OAVBBW=1.1*(AV2FX*AV1YI*(AV2FI*(.0753+(.442E-4)*PRES+(.649E-8)*PRES
1**2.0)-.0527*AV1YI)+(AV2FI**2.0)*AV2FX*(.0231+(.539E-4)*PRES+
2(.646E-8)*PRES**2.0)+AVDAX*AVDAI*(AV2FI*(.119+(.623E-4)*PRES+
3(.797E-8)*PRES**2.0)-.0787*AVDAI)+AV12I*AV12J*(PRES**0.5)*(AV12J
4*(PRES**0.5)*(1.107+(.2856E-4)*PRES)-.0835*AV12I))
OAVBBW=AVBBW+1.1*(.0861+AV2FX*
1AV2FJ**2.0+AVBB2*APPPK**2.0*(AV2FI*(.0398+(.12E-4)*PRES)+AV12I*
2(.0398+(.106E-4)*PRES)+2.32*AV12J+.00602*APPPK))
ORELIA=RELIAB+(1.4E-5)*PRES/AV12X+.0234/(AVBBW**0.333)+AIPA4*(1.0968+AIPA1*.250+AIPA2*.250+AIPA3*.150)+.11*ABFEW+.0136/ABFHJ+
2.0001135*VELA*TORQ/(PRES*ABFHW)+.208/(ABHBW**.333)
AIPBW=.0876*TRAL*AIPA4
AIPBR=AIPA4*TRAL*(.212+AIPA1*.338+AIPA2*.338+AIPA3*.274)
AFCCX=AAA5*(.00318*(VELL**2.)/APRHK*AAA8+TRAL+62.*AV12J)
AFCCI=AAA5*27.0*AV12J
RELIAB=RELIAB+.00196/AV12J*AAA5
AFCHX=AAA5*(2.0*AFCCX-62.0*AV12J)
ORELIA=RELIAB+.0385*AFCCI*AFCHX+.0301*AAA5*AIPBR+AAA5*(.00132/
11.0E-20+AFCCI))+AAA5*.07+.33+APRBR+.0487*APRBJ+.00743+.15*AVMAR
CALL OSWE (26.0*AV12J,1,RINGWT)
CALL OSFR (26.0*AV12J,0.0,6,1,RINGFR)
RELIAB=RELIAB+RINGFR
OATBBW=3.1E-5*FORC*(TRAL+.00318*AAA8*VELL**2./APRHK+AVBBX*AVBB2)
1+.2+.0907*APRBJ**2.0*(5.41*APRBJ+.735)
RELIAB=RELIAB+.1245/(ATBBW**.333)+14.0*ACBSR
OWEIGHT=WEIGHT+.2475*AFCHX*AFCCI**2.0+.268*AAA5*AIPBW+.704*AFCCI
1**3.0+AAA5*.0002+.0478+2.0*APRBW+.492*AVMAW+RINGWT+.252*AIPA4+
214.0*ACBSW+73.5*AFCCX*AV12J**2.0
CALL OSWE (ACBBY,1,RINGWT)
CALL OSFR (ACBBY,0.0,6,1,RINGFR)

```

TABLE IX (CONTINUED)

```

RELIAB=RELIAB+RINGFR+(AVDBR+AVDAR)*AAAA6
OALLX=(APRRI+APRBI)/2.0+1.836*APRBJ+APRHK+APPPX+(TRAL/2.0)+AFCCX
1+.00159*VELL**2.0/APRHK*AAAA8
ACTQM=FLOW+14.2*AV2FJ**3.0
ACTQL=(AV2FB+14.2*AV2FJ**3.0)*2.0
OWEIGHT=WEIGHT
1+14.*ACBSW+ACSOW+ACSCW+ABFAW+2.0*ABPBW+ABFW+ABFEW+AVDAW+AVBBW
2+AIPBW+ATBBW+ABH3W+AVDBW+RINGWT+(AVDBW+AVDAW)*AAAA6
ORELIAB=RELIAB
1+14.*ACBSR+ACSOR+ACSCR+ABPBR*2.0+AVDAR+AIPBR
722 PUNCH 830,PRES,ANOM,WEIGHT,RELIAB,OALLX,ACTQM,ACTQL
PUNCH 830,TRAL,VELL,APRHK,FORC,APPP1,ACBBI,APPPY
PUNCH 830,APPPX,APPK,APPPJ,AVBX,AAAA1,AAAA6,AAAA4
PUNCH 830,AKVEL,ANUMB,XMDC,XMDD,XHDB,XMDA,XMDE
PUNCH 830,AAAA2,AAAA3,AAAA5,AAA7,AAA8,AIPA1,AIPA2
PUNCH 830,AIPA3,FLOW,EINT,FLOA,AV2FJ,APRBJ,TORQ
PUNCH 830,AAA9,AAA10,ABFHJ,AFCCI,AKENG,XXXX1,VPRON
PUNCH 830,S1,S2,S3,S4,XXXX2,PPPP9
PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPMUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,TCST
GO TO 800
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
803 STOP
END

```

```

DIMENSION T1(12)
DIMENSION T2(12)
READ 820,T1(1),T1(2),T1(3),T1(4),T1(5),T1(6),T1(7)
READ 820,T1(8),T1(9),T1(10),T1(11),T1(12)
READ 820,T2(1),T2(2),T2(3),T2(4),T2(5),T2(6),T2(7)
READ 820,T2(8),T2(9),T2(10),T2(11),T2(12)
IF (SENSE SWITCH 1) 2050,800
2050 PRINT 2000
PRINT 2001
PRINT 2003
PRINT 2002
PRINT 2004
2000 FORMAT (47H          OUTPUT FROM PROGRAM 1    ACTUATOR-TRUSS)
2001 FORMAT (39H SYSTEM MOMENT ACTUATOR ACTUATOR)
2003 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
2002 FORMAT (40X,15H TRUSS TRUSS)
2004 FORMAT (40X,19H WEIGHT FAIL RATE)
800 READ 820,PRES,AMOM,ACTWT,ACTRB,ALLLX,ACTQM,ACTQL
READ 820,TRAL,VELL,APRHK,FORC,APPPI,ACBBI,APPPY
READ 820,APPPX,APPPK,APPPJ,AVBBX,AAAA1,AAAA6,AAAA4
READ 820,AKVEL,ANUMB,XMDC,XMDD,XMDB,XMDA,XMDE
READ 820,AAAA2,AAAA3,AAAA5,AAAA7,AAAA8,AIPA1,AIPA2
READ 820,AIPA3,FLOW,EINT,FLOA,AV2FJ,APRBJ,TORQ
READ 820,AAAA9,AAA10,ABFHJ,AFCCI,AKENG,XXXX1,VPRON
READ 820,S1,S2,S3,S4,XXXX2,PPPP9
READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
IF (AIPA1+AIPA2+AIPA3) 874,874,875
874 AIPA4=0.0
GO TO 876
875 AIPA4=1.0
876 ACTWT=.813*ACTWT+2.5
AV2FB=ACTOL/2.0-14.2*AV2FJ**3.0
VELA=VELL/AMOM
AV2F1=AV2FJ/.309
ACYCL=21.0E+8/(PRES*TRAL)
ALIFE=2.533*AV2F1*TORQ*VELA/(PRES*AV2FB**2.0)
ACTOA=ACTQM+.7*FLOW*(1.0-AAA1)
OAKINT=1.0/(.275*(TRAL+.00318*VELL**2.0/APRHK)/(2.4E+5*FORC/PRES)+1
APPP1**2.0*(ACBBI+APPPI)*(TRAL+APPPY+.00318*VELL**2.0/APRHK)/(14.8
2E+7*(FORC/PRES)**2.0*(ACBBI-APPPI))+(TRAL+APPPY+.00318*VELL**2.0/
3APRHK)/(3.14*(ACBBI**2.0-APPPY**2.0)**29.E+6))
OAKEXT=1.0/((APPPX-TRAL-.71*APPK**0.5-.00159*VELL**2.0/APRHK-APPPY
1)/(22.8E+6*(APPK**2.0*APPPJ**2.0))+(AVBBX+TRAL)/(2.37E+3*FORC))
OAKSS=(1.0-AAA1)*(1.0-AAA6)*1.0/(1.0/AKVEL*1(FORC/PRES)**2.0*PRES**0.5)+(AAA1)*(1.0-AAA4)*(1.0-AAA6)*1.0/
2(1.0/AKEXT+1.0/AKINT)+1.0E-9
FLOR=FLOW*(3000./PRES)**0.5
IF (FLOR-2.0) 871,871,872
871 AAA15=0.0
GO TO 873
872 AAA15=1.0

```

TABLE X
 . DECK NUMBER 4 - ACTUATOR & TRUSS
 COMPUTER PROGRAM LISTING

```

8730ABCST=5.20*(71.82*APPK-9.77*APPK**2.0+2.014*APPK**3.0+APPI**3.
1*(.875758-200.9068/PRES-79711.07/PRES**2.)+66628232./PRES**2.0-
2117434.89/PRES+105.55801-.034223256*PRES+1.0761896E-5*PRES**2.0)
3*(1.0+.0188*TRAL)+864.07+(556.78+2.293*FLOR-4.875E+14/(FLOR*
4.25974+133.3)**5.5629)*AAA15
OBCST=ABCST+(51.41+.0008176*PRES+.00646*PRES*APPI-20.285E-5*APPI
1**2.0*PRES-.6927*APPI+1.1711*APPI**2.0)*AAA8+AIPA4*(261.30+
227.30*AIPA3+35.10*(AIPA1+AIPA2)+31.20*TRAL*(AIPA1+AIPA2+AIPA3))
3+(100.0*AFCC1*1.00+250.0)*(1.0+.0188*TRAL)*(AAA2+.333*AAA6)
OBCST=ABCST+(167.49+.3822*FLOR-.8125E+14/(FLOR*.25974+133.3)**15.5629)*(AAA7+AAA1+0.5*AAA4)
OATCST=910.0+10.0*AAA8+AIPA4*(25.0+(AIPA1+AIPA2+AIPA3-1.0)*5.0)+1AAA2*10.0+AAA7*20.0+AAA6*20.0
OADCST=(138000.+1.E+4*AAA8+8000.*AIPA4+2.E+4*AAA2+14000.*AAA7
1+E+4*AAA6+1.E+4*(AAA1+AAA4)+18.0*ABCST)*AAA9+AAA10*154000.
AUCST=ABCST+ATCST
ACVOL=1.2*TRAL*FORC/PRES+ABFHJ**3.0*2.2
ACTIM=TRAL/VELL
XALFP=ATANF((AMOM-XMDC)/(XMDD-ALLX))
XBETP=ATANF((ALLX-XMDA)/(AMOM-XMDB))
XPHIP=1.57-XBETP
XSIGP=3.14-XALFP-XPHIP
XTHEP=ATANF((XMDE*COS(XBETP))/(2.0*(AMOM-XMDB)))
XMTLX=(XMDD-ALLX)/COS(XALFP)
XMTUX=XMDE/(2.0*SIN(XTHEP))
XMTUF=FORC*SIN(XALFP)/(2.0*SIN(XSIGP)*COS(XTHEP))
XMTLF=FORC*SIN(XPHIP)/SIN(XSIGP)
IF (ABSF(XMTLX)-ABSF(XMTUX))850,850,851
850 XMTPX=XMTUX
GO TO 852
851 XMTPX=XMTLX
852 XMRAX=XMTPX/64.34
IF (ABSF(XMTLF)-ABSF(XMTUF))853,853,854
853 XMTPF=XMTUF
GO TO 855
854 XMTPF=XMTLF
855 XTHIX=XMTPF/(216352.*XMRAX)
WEIGHT=1.777*XMRAX*XTHIX*(XMTLX+2.0*XMTUX)
RELIAB=3.0*(.00005/XTHIX+.00025/XMRAX)
WEIGHT=WEIGHT+6.338*APRBJ**3.0+.366*APRBJ**2.0
RELIAB=RELIAB+.0148950/APRBJ+.10
XRUWT=WEIGHT
WEIGHT=0.0
XRURB=RELIAB
RELIAB=0.0
XRHOP=ATANF((AMOM-XMDC)/(XMDD-ALLX-1.0))
XDELP=ATANF((AMOM-XMDB)/(ALLX-XMDA+1.0))
XZETP=ATANF((XMDE*(SIN(XDELP)))/(2.0*(AMOM-XMDB)))
XKTLX=(AMOM-XMDC)/(SIN(XRHOP))
XKTUX=XMDE/(2.0*(SIN(XZETP)))
OMTES=09.11E+7*XMRAX*XTHIX*((COS(XRHOP))*(ABSF(XMTLX-XKTLX))/XMTLX
1+2.0*(COS(XZETP))*(COS(XDELP))*(ABSF(XKTUX-XMTUX))/XMTUX)
AKSYS=1.0/(1.0/OMTES+1.0/AKINT+1.0/AKEXT)
ASTST=1.0/(1.0/AKSSS+1.0/OMTES)*(1.0-AAA6)
ANAFQ=AMOM*(AKSYS/EINT)**0.5
ACTWT=ACTWT+ACVOL*TOLW
XRUW1=XRUWT
XRUR1=XRURB
VFAC1=OMTES*(1.0-(VNAFQ/ANAFQ)**2.0)/AKSYS*(VNAFQ/ANAFQ)**2.0+1.0

```

```

IF(VFAC1-1.0)201,202,202
202 VFAC2=1.0
VFAC3=0.0
GO TO 200
201 IF(VFAC1-.333)203,204,204
204 VFAC2=VFAC1
VFAC3=1.0
GO TO 200
203 VFAC2=1.0
VFAC3=2.0
GO TO 200
200 VFAC4=1.0/VFAC2
XRUWT=VFAC4*XRUWT
XRURB=VFAC4*XURB
DO 902 N=1,11,1
IF (XTHIX-T1(N)) 901,901,902
902 CONTINUE
901 XCON1=T2(N)
XRUSU=(.8992*(2.0*XMRAX+XTHIX)**1.09885+.482)*(XMTLX+2.0*XMTUX)*(1
XCON1)+(37.75*APPPJ**1.5962+78.0)*4.0
XRUSD=(900.+12.50*(XMTLX+2.0*XMTUX)+10.*XRUSU)*XXXX1
ADTIM=AAAAA9*62.0+AAA10*21.0
IF (SENSE SWITCH 1) 722,723
722 PRINT 830,PRES,AMOM,ACTWT,ACTRB,XRUWT,XRURB
723 PUNCH 830,S1,S2,S3,S4,XXXX2,PPPP9,AAA10
PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,T0ILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3,XMDB,XMDC
GO TO 800
820 FORMAT(7E10.0)
830 FORMAT (7E10.3)
803 STOP
END
      .028      .035      .042      .049      .058      .065      .072
      .083      .095      .109      .120      10.0
10.15      10.50     11.00     11.60     12.10     12.87     13.44
14.23      15.20     17.30     19.61

```

TABLE X (CONTINUED)

```

Y=1.0
DIMENSION TUBE(9,14)
DIMENSION FIT(14)
DIMENSION T1(12)
DIMENSION T2(12)
READ 899,((TUBE(M,N),M=1,9),N=1,14)
READ 898,(FIT(N),N=1,14)
READ 820,T1(1),T1(2),T1(3),T1(4),T1(5),T1(6),T1(7)
READ 820,T1(8),T1(9),T1(10),T1(11),T1(12)
READ 820,T2(1),T2(2),T2(3),T2(4),T2(5),T2(6),T2(7)
READ 820,T2(8),T2(9),T2(10),T2(11),T2(12)
IF (SENSE SWITCH 1)2050,960
2050 PRINT 2000
PRINT 2001
PRINT 2002
2000 FORMAT (36H OUTPUT FROM PROGRAM 2 TUBE SYSTEM)
2001 FORMAT (38H SYSTEM MOMENT TUBING TUBING)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
960 READ 820,S1,S2,S3,S4,XXXX2,PPPP9,AAA10
READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPR,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTHT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLLX,XRUR1,XRUW1,VFAC3,XMDB,XMDC
997 TMTL1=S1
TMTL2=S3+S4+(AMOM-(XMDB+XMDC)/2.0)
FLOW=ACTQM
TMT2J=((1.02E-3*FLOW*(.794*TMTL1+TMTL2))/(.1*PRES))**.25
TMT1J=TMT2J*1.26
TMT2I=TMT2J*((1.0+6.83E-5*PRES)/(1.0-6.83E-5*PRES))**.5
TMT1I=TMT1J*((1.0+6.83E-5*PRES)/(1.0-6.83E-5*PRES))**.5
TMT2T=(TMT2I-TMT2J)/2.0
TMT1T=(TMT1I-TMT1J)/2.0
M=1
N=1
DO 901 N=1,14,1
IF (TMT2I-TUBE(M,N))902,902,901
901 CONTINUE
902 TMT2I=TUBE(M,N)
WTF2=FIT(N)
DO 903 M=2,9,1
IF (TMT2T-TUBE(M,N))904,904,903
903 CONTINUE
904 TMT2T=TUBE(M,N)
M=1
N=1
DO 911 N=1,14,1
IF (TMT1I-TUBE(M,N))912,912,911
911 CONTINUE
912 TMT1I=TUBE(M,N)
WTF1=FIT(N)

```

TABLE XI

DECK NUMBER 5 - TUBING
COMPUTER PROGRAM LISTING

```

DO 913 M=2,9,1
IF (TMT1T-TUBE(M,N))914,914,913
913 CONTINUE
914 TMT1T=TUBE(M,N)
TMT2J=TMT2I-2.0*TMT2T
TMT1J=TMT1I-2.0*TMT1T
OWEIGHT=.222*(TMTL1*(TMT1I**2.0-TMT1J**2.0)+2.0*TMTL2*
1(TMT2I**2.0-TMT2J**2.0))
ORELIAB=2.8E-6*((TMTL1*TMT1I)/TMT1T)+1.05E-6*(TMTL1/(TMT1I*TMT1T))
1+2.0*(2.8E-6*((TMTL2*TMT2I)/TMT2T)+1.05E-6*(TMTL2/(TMT2I*TMT2T)))
FVOL=.785*(TMTL1*(TMT1J**2.0)+2.0*TMTL2*(TMT2J**2.0))
TMTL3=S2
TMTL3W=.222*TMTL3*((TMT1I**2.0)-(TMT1J**2.0))
WEIGHT=WEIGHT+TMTL3W
ORELIAB=RELIAB+2.8E-6*((TMTL3*TMT1I)/TMT1T)+1.05E-6*
1(TMTL3/(TMT1I*TMT1T))
FVOL=FVOL+.785*(TMTL3*(TMT1J**2.0))
WEIGHT=WEIGHT+3.878E-2*FVOL
WTF3=WTF1
TMTFW1=WTF1*2.0
TMTFW2=WTF2*8.0
TMTFW3=WTF3*2.0
WEIGHT=WEIGHT+TMTFW1+TMTFW2+TMTFW3
TMTFR1=.06*(TMT1I)+1.47E-3/(TMT1I*TMT1T)
TMTFR2=.06*(TMT2I)+1.47E-3/(TMT2I*TMT2T)
TMTFR3=.06*(TMT1I)+1.47E-3/(TMT1I*TMT1T)
TMTFR=2.0*TMTFR1+8.0*TMTFR2+2.0*TMTFR3
RELIAB=RELIAB+TMTFR
WEIGHT=WEIGHT*2.0
RELIAB=RELIAB*2.0
TWGHT=WEIGHT
TFAIL=RELIAB
DO 702 N=1,11,1
IF (TMT1T-T1(N)) 701,701,702
702 CONTINUE
701 TCON1=T2(N)
DO 703 N=1,11,1
IF (TMT2T-T1(N)) 703,703,704
704 CONTINUE
703 TCON2=T2(N)
OTUC SU=(120.0+2.3*ACTWT**0.5)*ANUMB+(.8992*(TMT1I)**1.09885+.482)*(
1(TMTL1)*(TCON1)+(.8992*(TMT2I)**1.09885+.428)*(TMTL2)*(TCON2))*(
2ANUMB/2.0)
OTUC SD=(450.0+12.50*(ANUMB/2.0)*(TMTL1+TMTL2))*(XXXX2)+(460.0)*
1ANUMB*AAA10+460.0*S7*PPPP3+460.0*S8*PPPP4+460.0*RRRR4+120.0*RRRR4
2*SSSI+230.0*FFFF1*FFFF4
IF (SENSE SWITCH 1) 821,822
821 PRINT 830,PRES,AMOM,TWGHT,TFAIL
822 PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLX,XRUR1,XRUW1,VFAC3,PPPP9

```

TABLE XI (CONTINUED)

PUNCH 830,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMTII
 IF (Y-3.0)950,950,951

950 Y=Y+.00001
 GO TO 960

820 FORMAT (7E10.0)
 830 FORMAT (7E10.3)
 899 FORMAT (9E8.0)
 898 FORMAT (7E10.0)

951 STOP
 END

| | 2.8E-2 | 3.2E-2 | 3.5E-2 | 4.2E-2 | 0.0E+0 | 0.0E+0 | 0.0E+0 | 0.0E+0 |
|----------|---------|---------|---------|---------|---------|----------|---------|---------|
| 1.25E-1 | 2.8E-2 | 3.2E-2 | 3.5E-2 | 4.2E-2 | 0.0E+0 | 0.0E+0 | 0.0E+0 | 0.0E+0 |
| 1.88E-1 | 3.2E-2 | 3.5E-2 | 4.2E-2 | 0.0E+0 | 0.0E+0 | 0.0E+0 | 0.0E+0 | 0.0E+0 |
| 2.5E-1 | 3.5E-2 | 4.2E-2 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 0.0E+0 | 0.0E+0 |
| 3.13E-1 | 3.5E-2 | 4.2E-2 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 0.0E+0 | 0.0E+0 |
| 3.75E-1 | 3.5E-2 | 4.2E-2 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 0.0E+0 | 0.0E+0 |
| 5.0E-1 | 3.5E-2 | 4.2E-2 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 |
| 6.25E-1 | 3.5E-2 | 4.2E-2 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 |
| 7.5E-1 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 |
| 8.75E-1 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 |
| 1.0E+0 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 |
| 1.25E+0 | 4.9E-2 | 5.8E-2 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 |
| 1.5E+0 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 | 1.34E-1 | 0.0E+0 |
| 1.75E+0 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 | 1.34E-1 | 0.0E+0 |
| 2.0E+0 | 6.5E-2 | 7.2E-2 | 8.3E-2 | 9.5E-2 | 1.09E-1 | 1.25E-1 | 1.34E-1 | 0.0E+0 |
| 1.53E-2 | 1.95E-2 | 2.4E-2 | 3.45E-2 | 4.39E-2 | 6.72E-2 | 1.058E-1 | | |
| 1.585E-1 | 2.02E-1 | 2.43E-1 | 3.35E-1 | 4.25E-1 | 5.1E-1 | 6.0E-1 | | |
| .028 | .035 | .042 | .049 | .058 | .065 | .072 | | |
| .083 | .095 | .109 | .120 | 10.0 | | | | |
| 10.15 | 10.50 | 11.00 | 11.60 | 12.10 | 12.87 | 13.44 | | |
| 14.23 | 15.20 | 17.30 | 19.61 | 19.61 | | | | |

TABLE XI (CONTINUED)

```

DIMENSION BRI(5,22)
DIMENSION BTI(5,18)
DIMENSION BTII(4,18)
READ 103,((BRI(M,N),M=1,5),N=1,22)
READ 104,((BTI(K,L),K=1,5),L=1,18)
READ 105,((BTII(I,J),I=1,4),J=1,18)
X=1.0
200 READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRDN
READ 820,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
READ 820,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
IF (S7)931,931,932
932 PUMS=PUMS1
ANGL=ANGL1
FLOW=ANUMB*ACTQA*PPPP1
PAPW1=1.0
1020 PAPWP=((8.49E-3*(PAPW1**2.0))/(7.08-5.3E-5*PRES))+(14.88E-6*
1PRES)/(PUMS*(SINF(ANGL)/COSF(ANGL))*(2.98+2.44E-4*PRES)*(7.08-
25.3E-5*PRES))+{FLOW/(PUMS*(SINF(ANGL)/COSF(ANGL))*(2.98+2.44E-4
3*PRES)*(7.08-5.3E-5*PRES))}+((1.24E-4*PRES*PAPW1)/(PUMS*
4(SINF(ANGL)/COSF(ANGL))*(7.8-5.3E-5*PRES)))*#.333
IF (ABSF(PAPWP-PAPW1)-1.0E-5*PAPWP)100,100,101
101 PAPW1=(PAPWP+PAPW1)/2.0
GO TO 102
100 PAPWI=PAPWP*(3.98+4.14E-4*PRES)
PAPWJ=PAPWP*(2.98+2.44E-4*PRES)
PAPWK=PAPWP*(1.98+7.4E-5*PRES)
PAPWX=2.71*PAPWP*(SINF(ANGL)/COSF(ANGL))*(2.98+2.44E-4*PRES)
WT=.21*PAPWX*((PAPWI**2.0)-9.0*(PAPWP**2.0)-(PAPWK**2.0))
PAPEW=.181*(PAPWP**3.0)
OPAPER=(5.25E-7*PAPEW*(PAPWJ**2.0)*(PUMS**2.0)*PRES*(SINF(ANGL)/
1COSF(ANGL))*(2.98+2.44E-4*PRES))/(FLOW*PAPWP)
REL=9.0*PAPER+(9.8E-2*PAPER)+((1.98E-5*PRES)/PAPWJ)
PAPDX=1.51*PAPWP*(SINF(ANGL)/COSF(ANGL))*(2.98+2.44E-4*PRES)
OWT=WT+9.0*PAPEW+(.863*(PAPWP**3.0)*(.249+2.74E-5*PRES*
1(SINF(ANGL)/COSF(ANGL))*(2.98+2.44E-4*PRES)))
OPAPC1=.122*PAPWP*((2.98+2.44E-4*PRES)*(SINF(ANGL)/COSF(ANGL))*1
1PRES)**.333
PAPCZ=2.05*PAPWK
PAPCJ=PAPWP*(3.24+2.44E-4*PRES)
PAPCI=PAPC1+.20
PAPFI=1.26*PAPWP
PAPT1=PAPCJ+PAPFI+1.7E-4*PAPWP*PRES
M=1
N=1
DO 108 N=1,22,1
IF (PAPT1-BRI(M,N))107,107,108
108 CONTINUE
107 PAPTK=BRI(2,N)

```

TABLE XII

DECK NUMBER 6 - FIXED ANGLE PUMP
COMPUTER PROGRAM LISTING

```

PAPTI=BRI(3,N)
PAPTY=BRI(4,N)
PAPTW=BRI(5,N)
K=1
L=1
DO 109 L=1,18,1
IF (PAPCI-BTI(K,L))110,110,109
109 CONTINUE
110 PAPGK=BTI(2,L)
PAPGI=BTI(3,L)
PAPGY=BTI(4,L)
PAPGW=BTI(5,L)
I=1
J=1
DO 111 J=1,18,1
IF (PAPCI-BTII(I,J))112,112,111
111 CONTINUE
112 PAPUI=BTII(2,J)
PAPUY=BTII(3,J)
PAPUW=BTII(4,J)
GO TO 933
931 PAPWP=0.0
REL=0.0
WT=0.0
PAPC1=0.0
PAPT1=0.0
PAPFI=0.0
PAPCI=0.0
PAPCJ=0.0
PAPCZ=0.0
PAPDX=0.0
PAPER=0.0
PAPWX=0.0
PAPWK=0.0
PAPWI=0.0
PAPWJ=0.0
PAPTK=0.0
PAPTI=0.0
PAPTY=0.0
PAPTW=0.0
PAPGK=0.0
PAPGI=0.0
PAPGY=0.0
PAPGW=0.0
PAPUI=0.0
PAPUY=0.0
PAPUW=0.0
933 PUNCH 151,PAPWP,PAPWI,PAPWJ,PAPWK,PAPWX
PUNCH 151,PAPER,PAPDX,PAPCZ,PAPCJ,PAPCI
PUNCH 151,PAPFI,PAPTI,PAPC1,WT,REL,PAPUW
PUNCH 151,PAPTK,PAPTI,PAPTY,PAPTW,PAPGK
PUNCH 151,PAPGI,PAPGY,PAPGW,PAPUI,PAPUY
PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPR,VTEST,VTCST

```

TABLE XII (CONTINUED)

```

PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTOA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
PUNCH 830,FVOL,TWHT,TFAIL,TUCSU,TUCSD,TMTII
IF (X-10.0)203,203,204
203 X=X+.00001
GO TO 200
103 FORMAT (5E12.0)
104 FORMAT (5E12.0)
105 FORMAT (4E12.0)
151 FORMAT (5E13.0)
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
204 STOP
END
    7.87E-1    7.87E-1    1.457E+0    3.54E-1    9.0E-2
    9.84E-1    9.84E-1    1.654E+0    3.54E-1    1.0E-1
   1.181E+0    1.181E+0    1.850E+0    3.54E-1    1.2E-1
   1.378E+0    1.375E+0    2.175E+0    3.94E-1    1.9E-1
   1.575E+0    1.575E+0    2.441E+0    4.72E-1    2.9E-1
   1.772E+0    1.772E+0    2.677E+0    4.72E-1    3.4E-1
   1.969E+0    1.969E+0    2.835E+0    4.72E-1    3.5E-1
   2.165E+0    2.165E+0    3.150E+0    5.12E-1    4.7E-1
   2.362E+0    2.362E+0    3.347E+0    5.12E-1    5.1E-1
   2.559E+0    2.559E+0    3.543E+0    5.12E-1    5.4E-1
   2.756E+0    2.756E+0    3.937E+0    6.30E-1    8.6E-1
   2.953E+0    2.953E+0    4.134E+0    6.30E-1    1.04E+0
   3.150E+0    3.150E+0    4.331E+0    6.30E-1    1.10E+0
   3.347E+0    3.347E+0    4.724E+0    7.09E-1    1.41E+0
   3.543E+0    3.543E+0    4.921E+0    7.09E-1    1.48E+0
   3.740E+0    3.740E+0    5.118E+0    7.09E-1    1.53E+0
   3.937E+0    3.937E+0    5.512E+0    7.87E-1    2.04E+0
   4.134E+0    4.134E+0    5.709E+0    7.87E-1    2.12E+0
   4.331E+0    4.331E+0    5.906E+0    7.87E-1    2.20E+0
   4.724E+0    4.724E+0    6.496E+0    8.66E-1    3.02E+0
   5.118E+0    5.118E+0    7.087E+0    9.45E-1    4.02E+0
   5.512E+0    5.512E+0    7.480E+0    9.45E-1    4.38E+0
   4.724E-1    4.724E-1    1.457E+0    4.724E-1    1.4E-1
   5.906E-1    5.906E-1    1.654E+0    5.118E-1    1.9E-1
   6.693E-1    6.693E-1    1.851E+0    5.512E-1    2.6E-1
   7.874E-1    7.874E-1    2.047E+0    5.906E-1    3.3E-1
   9.843E-1    9.843E-1    2.441E+0    6.493E-1    5.1E-1
   1.181E+0    1.181E+0    2.635E+0    7.480E-1    7.8E-1
   1.378E+0    1.378E+0    3.150E+0    8.268E-1    1.00E+0
   1.578E+0    1.578E+0    3.543E+0    9.055E-1    1.47E+0
   1.772E+0    1.772E+0    3.937E+0    9.843E-1    1.97E+0
   1.969E+0    1.969E+0    4.331E+0    1.063E+0    2.56E+0
   2.165E+0    2.165E+0    4.724E+0    1.142E+0    3.20E+0
   2.362E+0    2.362E+0    5.118E+0    1.221E+0    4.03E+0
   2.560E+0    2.560E+0    5.512E+0    1.299E+0    5.05E+0
   2.756E+0    2.756E+0    5.906E+0    1.378E+0    5.96E+0
   2.963E+0    2.963E+0    6.299E+0    1.457E+0    7.41E+0
   3.150E+0    3.150E+0    6.693E+0    1.535E+0    8.79E+0
   3.347E+0    3.347E+0    7.087E+0    1.614E+0    1.04E+1
   3.543E+0    3.543E+0    7.480E+0    1.693E+0    1.21E+1
   4.724E-1    1.103E+0    3.15E-1     4.0E-2
   5.906E-1    1.260E+0    3.54E-1     6.0E-2
   6.693E-1    1.378E+0    3.94E-1     1.3E-1

```

TABLE XII (CONTINUED)

| | | | |
|----------|----------|---------|---------|
| 7.874E-1 | 1.654E+0 | 4.72E-1 | 1.6E-1 |
| 9.843E-1 | 1.850E+0 | 4.72E-1 | 2.5E-1 |
| 1.181E+0 | 2.165E+0 | 5.12E-1 | 3.4E-1 |
| 1.378E+0 | 2.441E+0 | 5.51E-1 | 5.0E-1 |
| 1.578E+0 | 2.677E+0 | 5.91E-1 | 6.3E-1 |
| 1.772E+0 | 2.953E+0 | 6.30E-1 | 6.9E-1 |
| 1.969E+0 | 3.150E+0 | 6.30E-1 | 7.5E-1 |
| 2.165E+0 | 3.543E+0 | 7.09E-1 | 1.00E+0 |
| 2.362E+0 | 3.740E+0 | 7.09E-1 | 1.25E+0 |
| 2.559E+0 | 3.937E+0 | 7.09E-1 | 1.50E+0 |
| 2.756E+0 | 4.331E+0 | 7.87E-1 | 1.56E+0 |
| 2.953E+0 | 4.528E+0 | 7.87E-1 | 2.13E+0 |
| 3.150E+0 | 4.921E+0 | 8.66E-1 | 2.25E+0 |
| 3.347E+0 | 5.118E+0 | 8.66E-1 | 2.88E+0 |
| 3.543E+0 | 5.512E+0 | 9.45E-1 | 3.50E+0 |

TABLE XII (CONTINUED)

```

X=1.0
IF (SENSE SWITCH 1)2050,302
2050 PRINT 2000
PRINT 2001
PRINT 2002
2000 FORMAT (41H OUTPUT FROM PROGRAM 3 FIXED ANGLE PUMP)
2001 FORMAT (37H SYSTEM MOMENT PUMP PUMP)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
302 READ 151,PAPWP,PAPWI,PAPWJ,PAPWK,PAPWX
READ 151,PAPER,PAPDX,PAPCZ,PAPCJ,PAPCI
READ 151,PAPFI,PAPTI,PAPC1,WT,REL,PAPUW
READ 151,PAPTK,PAPTI,PAPTY,PAPTW,PAPGK
READ 151,PAPGI,PAPGY,PAPGW,PAPUI,PAPUY
READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIPP,VLIFA,VCYCA,VOPER,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLX,XRUR1,XRUW1,VFAC3,PPPP9
READ 820,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMTII
IF (S7) 931,931,932
932 PUMS=PUMS1
FLOW=ANUMB*ACTQA*PPPP1
ANGL=ANGL1
TANA=SINF(ANGL)/COSF(ANGL)
PAPCX=1.135*(PAPCZ+1.35*PAPC1-(PAPTY+.050))+.33
OPAPCW=.224*((PAPTY+.05)*(PAPTK**2.0)+PAPCX*(PAPGK**2.0)-9.0*
1(PAPFI**2.0)*(PAPTY+.05)-2.05*(PAPWK**3.0)-
2(PAPCI**2.0)*(1.35*PAPC1+.33))
WT=WT+PAPCW+PAPUW+PAPTW+PAPGW
OREL=REL+1.81E-4/(PAPWP**2.0)+1.23E-2/PAPC1+.099*((PUMS/60.0)**.33)
1+.231*((PUMS/60.0)**.33)
PAPKI=1.8E-3*((FLOW*PRES/PUMS)**.33)
PAPMI=.0032*((FLOW*PRES)/(PUMS*PAPKI))**0.5
PAPHI=.214*PAPWK
OWT=WT+9.0*(.184*(PAPTY+.05)*(PAPFI**2.0)-.061*(PAPWP**3.0))+.276*
1PAPDX*(PAPKI**2.0)+4.8E-4*(FLOW*PRES/PUMS)+1.46E-2*(PAPWK**3.0)
REL=REL+8.1E-3/(PAPWP**2.0)+3.6E-4/PAPKI+1.2E-2
PAPIY=.5*PAPWK
PAPIK=.68*PAPWK
PAPNI=.81*PAPWK
OWT=WT+.0715*(PAPWK**2.0)*PAPCZ-.069*(PAPHI**2.0)*PAPWK+1.36*(1.57*-
1(PAPNI**2.0)*PAPIY-PAPNI*PAPKI*PAPIY-1.57*(PAPMI**2.0)*(PAPNI-
2PAPKI))+PAPIY*(.062*(PAPWK**2.0)+.054*PAPCZ*PAPIK)
REL=REL+8.4E-4/PAPWK+.8E-4/PAPNI+2.72E-3/PAPNI
OWT=WT+.042*(PAPIK**4.0)+2.62E-4*PAPWK+.0113*(PAPWK**2.0)*
1(2.21*PAPWX+1.0)+.0148*PAPWX*(PAPWK**2.0)
REL=REL+.021+4.0E-3+8.4E-4/PAPWK+(1.2E-3*PUMS)/PAPWK
PAPQI=.32E-2*PAPWP*(PRES**.5)
PAPSI=PAPQI+.125
PAVII=1.65*PAPSI
OWT=WT+3.46*(PAPQI**3.0)+.019*(PAPWX**3.0)+.216*(PAPQI**2.0)+.0148

```

TABLE XIII

DECK NUMBER 7 - FIXED ANGLE PUMP

COMPUTER PROGRAM LISTING

```

1*PAPQI-.048*PAPWX*(PAPQI**2.0)+.872*(PAPQI**3.0)+.155*(PAPSI**3.0)
REL=REL+7.0E-3/PAPQI+3.6E-2/PAPWX+3.65E-4/(PAPQI**2.0)+1.5E-3
CALL OSWE (PAVII,1,RINGWT)
WT=WT+RINGWT
CALL OSFR (PAVII,.0133*PRES,5,1,RINGFR)
REL=REL+RINGFR
PAHII=PAPTI+.12
OWT=WT+4.28E-3*(PAPCI**2.0)+5.26E-4*PAPCI+8.9E-6+7.68E-3*
1(.366*(PAPTI**2.0)+8.78E-2*PAPTI+5.26E-3)+1.17E-2*(PAPGI**2.0)-
25.16E-3*PAPGI*PAPUI-6.54E-3*(PAPUI**2.0)
REL=REL+5.0E-4+6.0E-4+6.0E-4
CALL OSWE (PAPCI,1,RINGWT)
WT=WT+RINGWT
CALL OSFR (PAPCI,.0133*PRES,5,1,RINGFR)
REL=REL+RINGFR
PAFCI=PAPCI+.42
PAFP1=PAFCI+.04
PAFPK=1.24*PAPCI+9.92E-2
PAFPX=.216*(PAPCI**.5)
PAFP2=.393*(PAPCI**.5)
CALL OSWE (PAFPK,1,RINGWT)
CALL OSFR (PAFPK,.0133*PRES,5,1,RINGFR)
OWT=WT+6.96E-2*(PAPCI**2.0)+1.46E-2*PAPCI+1.15E-2*(PAPCI**.5)*
1(PAFPI**2.0)+RINGWT
REL=REL+3.45E-2*PAPCI+.4*PAFCI+RINGFR
PAFYI=PAFP1+.05
OWT=WT+1.3E-2*PAFPX*PAFYI+1.0E-4+9.8E-4*PAFPK+2.29E-3*PAFPK+3.21E-4
1+1.3E-3*PAPCI-3.0E-5+4.64E-3*PAPCI+6.95E-5+2.31E-2*PAFYI*PAFPX+
2(3.46E-2*(PAFYI**2.0)+.18*PAFYI+3.51E-2)*(PAFP2+.07)+
32.16E-2*PAFYI+1.27E-2
REL=REL+1.0E-3+6.25E-2+6.0E-4+5.0E-4+1.0E-2+1.39E-2*PAFYI
CALL OSWE (PAFYI+.1,1,RINGWT)
CALL OSFR (PAFYI+.1,.0133*PRES,5,1,RINGFR)
PAFMI=.3125
WT=WT+RINGWT+6.6E-3
REL=REL+RINGFR+1.0E-3
CALL OSWE (PAFMI,1,RINGWT)
CALL OSFR (PAFMI,.0133*PRES,5,1,RINGFR)
PAFKI=1.36*PAPGI
OWT=WT+RINGWT+.0278*(PAPGI**3.0)+.177*(PAPUI**2.0)*(PAPUY+
1.51*PAPCI)+.0142*PAPTY*(PAPTI**2.0)
REL=REL+RINGFR+.049/PAHII+.053/PAPUI
CALL OSWE (PAHII,2,RINGWT)
CALL OSFR (PAHII,.0133*PRES,5,2,RINGFR)
PARDI=2.0*PAPWJ
PARGK=1.08*PARDI
PARGI=PARGK*(1.0+1.09E-4*PRES)
PARAX=.19*PAPWI
PARMI=9.25E-3*((FLOW*(PRES**0.5))**0.5)
OWT=WT+RINGWT+.346*(PAPWJ**2.0)-.506*PAPWJ*PAPWP+.092*(PARGI**2.0)
1-.0722*(PARGK**2.0)+6.4E-4*(PARDI**2.0)+.0453*PAPWI*(PARGI**2.0)
2-.0398*(PAPWI**3.0)+.705*(PARMI**3.0)
REL=REL+RINGFR+.155/(PAPWJ**2.0)+4.0E-4+.364/PARDI+2.8E-2+1.3E-3
CALL OSWE (PARMI,1,RINGWT)
WT=WT+RINGWT
CALL OSFR (PARMI,.0133*PRES,5,1,RINGFR)
REL=REL+RINGFR
CALL OSWE (PARGK+.1,1,RINGWT)
WT=WT+3.0*RINGWT

```

TABLE XIII (CONTINUED)

```

CALL OSFR (PARGK+.1,PRES,6,1,RINGFR)
REL=REL+3.0*RINGFR
PACLP=(.462*PARDI)/(PRES**.25)
PACLK=S5*(1.333*PACLP)
PACLW=S5*(22.5*(PACLP**3.0))
PACLR=S5*((2.91E-5*(PRES**.5))/(PACLP**2.0))
PACKW=S5*.885*(PACLK**3.0)
PACKR=PACLR
PACBI=4.66*PACLK
PACBX=18.0*PACLK
PACBW=72.0*(PACLK**3.0)
PACBR=S5*.21
WT=WT+PACLW+PACKW+PACBW
REL=REL+PACLR+PACKR+PACBR
PARJK=1.7*PAPWJ
PARJI=.0935*(PARJK**.667)
OWT=WT+.0139*(PARJK**2.334)+.0924*((PARGI**2.0)-1.13*(PAPWI**2.0))
1*(PAPWX+PAPDX-PARAX)+.266*PAPTY*((PAFKI**2.0)-(PAHII**2.0))
REL=REL+1.24E-2/(PARJI**2.0)+.122/PAPWI
PACMK=1.09*((FLOW**.5)/(PRES**.25))
CALL OSWE (PACMK,1,RINGWT)
CALL OSFR (PACMK,PRES,5,1,RINGFR)
OWT=WT+RINGWT+(.0645*(PARGI**2.0)-.015*(PAPWJ**2.0))*258*PARJI+
1.115*PACBX*(PACBI**2.0)*S5+.134*PARGI*(FLOW/(PRES**.5))+2.43*(PACMK**3.0)
REL=REL+RINGFR+3.78E-2/PACMK+8.0E-3/PAPWJ+3.1E-2/PACMK
PADS=PAPWP**3.0*TANA*(21.05+(1.72E-3*PRES))
PADS1=23.4*PADS
WT=WT+TOILW*PADS1
PUWT1=WT
PREL1=REL
PALF=10425.0/PUMS
250 IF (PADS-1.4) 251,252,252
251 S10=0.0
GO TO 253
252 S10=1.0
253OPFAVU=S5*(1.0-S10)*(160.85*(PADS+.054)**1.582+1494.44)+S5*S10*(1471.38+612.51*PADS+154.43/PADS**3.0+400.)+(1.0-S5)*(1.0-S10)*2128.68*((PADS+.054)**1.582+875.55+380.0)+(1.0-S5)*S10*(377.10+3490.10*PADS+123.54/PADS**3.0+380.0)
OPFAVD=S5*((85000.0+18.*PFAVU)*PPPP2+93000.0*PPPP3)+(1.0-S5)*((165000.0+18.0*PFAVU)*PPPP2+74000.0*PPPP3)
PFAVT=48.0*PPPP2+19.0*PPPP3+S5*(6.0*PPPP2+4.0*PPPP3)
PAOP=0.0
PFAFD=PFAVD
PFAFT=PFAVT
PFAFU=PFAVU
GO TO 933
931 PUMS=0.0
REL=0.0
WT=0.0
ANGL=0.0
PUWT1=0.0
PREL1=0.0
PALF=0.0
PAOP=0.0
PADS1=0.0
PFAFD=0.0
PFAFT=0.0

```

TABLE XIII (CONTINUED)

```
PFAFU=0.0
933 IF (SENSE SWITCH 1) 921,922
921 PRINT 830,PRES,AMOM,WT,REL
922 PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
PUNCH 830,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
PUNCH 830,PUWT1,PREL1,PALF,PAOP,PADS1,PFAFD,PFAFT
PUNCH 830,PFAFU
IF (X-10.0)300,300,301
300 X=X+.00001
GO TO 302
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
151 FORMAT (5E13.0)
301 STOP
END
```

TABLE XIII (CONTINUED)

```

IF (SENSE SWITCH 1)2050,499
2050 PRINT 2000
PRINT 2001
PRINT 2002
2000 FORMAT (42H OUTPUT FROM PROGRAM 4 WOBBLE PLATE PUMP)
2001 FORMAT (37H SYSTEM MOMENT PUMP PUMP)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
499 READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
READ 820,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
READ 820,PUWT1,PREL1,PALF,PAOP,PADS1,PFAFD,PFAFT
READ 820,PFAFU
X=1.0
IF(S8)931,931,932
932 PUMS=PUMS2
ANGL=ANGL2
FLOW=ANUMB*ACTQA*PPPP9
TANA=SINF(ANGL)/COSF(ANGL)
PWBB1=0.1
5030 PWBBP=(FLOW/(PUMS*TANA*(20.5-1.92E-4*PRES))-  

1(((PWBB1**2.0)*(3.58-3.46E-5*PRES))/(20.5-1.92E-4*PRES))+  

2((4.44E-3*PWBB1)/(20.5-1.92E-4*PRES))+  

3((1.64E-5*PRES*PWBB1)/((1.45*PWBB1+.26)*(TANA**2.0)*PUMS*  

4(20.5-1.92E-4*PRES))))**.333
IF (ABSF(PWBBP-PWBB1)-1.0E-5*PWBBP)501,501,502
502 PWBB1=(PWBBP+PWBB1)/2.0
GO TO 503
501 PWBB1=PWBBP*(3.98+2.46E-4*PRES)+.523
PWBBK=.84*PWBBP
PWBBX=5.57*TANA*(2.9*PWBBP+.523)
PWBBJ=2.9*PWBBP+.523
PWBBR=(1.74E-5*PWBBP*TANA*(PRES**2.0))/FLOW
OWT=.199*((((PWBB1**2.0)-.706*(PWBBP**2.0))*PWBBX-(95.5*(PWBBP**3.0)  

1-13.5*(PWBBP**2.0)-3.08*PWBBP+.438)*TANA-9.0*PWBBX*(PWBBP**2.0))
REL=PWBBR
CALL OSWE (PWBBK,2,RINGWT)
WT=WT+RINGWT
CALL OSFR (PWBBK,PRES,5,2,RINGFR)
REL=REL+RINGFR
PWPEI=1.535E-2*PWBBP*(PRES**.5)
PWPER=(1.42E-5*PUMS)/(PWBBP)
OWT=WT+9.0*.95E-6*(PWBBP**3.0)*(PRES**1.5)+9.0*(.176*PWBBX*  

1(PWBBP**2.0)+.248*(PWPEI**3.0))+S6*9.0*(TANA*(.206*(PWBBP**3.0)+  

2.0372*(PWBBP**2.0)))+S6*(9.77E-3*PWBBP*TANA*(PWBBJ**3.0))/PWBBK
OREL=REL+9.0*PWPER+9.0*(3.8E-3*PWBBR+.9*PWPER)+S6*9.0*  

1(1.27E-2*PWBBR)+S6*.05
PWBNP=9.13E-3*PUMS*PWBBP*(PWBBJ**.5)
PWBNR=(8.15E-4*PUMS)/(PWBNP)

```

TABLE XIV

DECK NUMBER 8 - WOBBLE PLATE PUMP
COMPUTER PROGRAM LISTING

```

PWPJI=1.333*PWBBJ
PWPJY=.0452*PWBBP*TANA*(PRES**.5)
PWPJR=(.005*PUMS)/(PWBBJ)
OWT=WT+9.0*(6.5E-2*(PWPEI**3.0))+.617*PWBBJ*TANA*(PWBNP**2.0)+1.0715*PWBNP*(PWBBJ**2.0)+.27*PWBBJ*PWBNP*PWPEI+.428*(PWBNP**3.0)
2+.705*PWBNP*(PWPEI**2.0)+.393*PWPJY*(PWBBJ**2.0)
REL=REL+9.0*(.667*PWPER)+PWBNR+2.0*PWBNR+PWPJR
PWPG1=((3.32E-3*(PWBBP**3.0)+6.0E-4*(PWBBP**2.0))*TANA*PRES)**.333
PWPGP=PWPG1+.225
PWPPP=1.133*PWPGP
CALL OSWE (PWPPP,2,RINGWT)
OWT=WT+.103*TANA*(PWPJI**3.0)+.054*(PWPJI**2.0)*(PWPJY+.33)+1.191*(PWPGP**3.0)-.377*(PWPG1**3.0)+2.77E-2*(PWPGP**3.0)+2.0452*(PWPGP**3.0)+RINGWT
CALL OSFR (PWPPP,.0133*PRES,5,2,RINGFR)
REL=REL+2.0*PWPER+.8*PWPJR+.1333*PWPGP+8.9E-2*PWPGP+RINGFR
CALL OSWE (PWPGP,2,RINGWT)
CALL OSFR (PWPGP,.0133*PRES,5,2,RINGFR)
PWPPIR=(2.66*PUMS*TANA)/(PRES*(PWBBP))
OWT=WT+RINGWT+(4.23E-5*PRES*(PWBBP**2.0))/TANA+11.57E-2*(PWPJI**2.0)-(4.65E-6*PRES*(PWBBP**2.0))/TANA+25.68E-2*(PWPGP**2.0)+3.32E-3*PWPGP+1.38E-2*(PWPGP**2.0)+3S6*(2.0E-3*(FLOW**1.5))/(PRES**.75)+S6*5.27E-2*PWBBX*(PWBBK**2.0)
OREL=REL+RINGFR+RINGFR+PWPPIR+4.0*PWPPIR+1.33E-2*PWPGP+11.775E-2*PWPGP+5.62E-2/PWPGP+S6*(9.75E-2*(FLOW**.5))/(PRES**.25)
2+S6*1.25E-2/PWBBK
CALL OSWE (PWBBK,1,RINGWT)
CALL OSFR (PWBBK,PRES,5,1,RINGFR)
PWCHI=1.25E-2*(PWBBJ**.333)*(PWBBK**.667)*(PRES**.333)
PWCII=1.225*PWBBJ
PWCIR=.1125/PWBBJ
OWT=WT+S6*RINGWT+S6*(6.67E-4*(PWBBJ**1.67)*(PWBBK**1.333)*1(PRES**.667)+97.0*(PWCHI**3.0)+.546*PWCHI*(PWBBJ**2.0))
OREL=REL+S6*RINGFR+S6*(4.05E-3/(PWCHI**2.0)+(1.333E-4*(PWBBJ**2.0))/1/(PWCHI**3.0)+PWCIR)
CALL OSWE (PWCII,2,RINGWT)
CALL OSFR (PWCII,PRES,5,2,RINGFR)
PWCLI=.189*PWBBJ
PWCLW=4.69E-2*PWCHI*(PWBBJ**2.0)
PWCLR=3.375E-2/PWBBJ
WT=WT+S6*(RINGWT+4.06*(PWCHI**3.0)+PWCLW)
REL=REL+S6*(RINGFR+.014+PWCLR)
CALL OSWE (PWCLI,2,RINGWT)
CALL OSFR (PWCLI,PRES,5,2,RINGFR)
WT=WT+S6*(RINGWT+.147*PWCLW+5.75E-2*PWBBX*(PWBBK**2.0))
PWCBR=.0125/PWBBK
REL=REL+S6*(RINGFR+.667*PWCLR+2.0*PWCBR)
CALL OSWE (PWBBI,2,RINGWT)
CALL OSFR (PWBBI,PRES,5,2,RINGFR)
PWVHI=PWBBI*(1.0+1.516E-4*PRES)
PWHFI=9.11E-2*(FLOW**0.5)
OWT=WT+3.0*RINGWT+.394*RINGWT+6.1E-1*(PWBBP**3.0)+(.15*(FLOW**1.5))/1/(PRES**.75)+2.3E-2*(PWCII**3.0)+PRES*(PWBBI**3.0)*(1.03E-5+21.875E-9*PRES+1.1E-13*(PRES**2.0))+8.6E-4*(PWBBP**3.0)*3(PRES**.667)+2.34E-6*(FLOW**1.5)
OREL=REL+3.0*RINGFR+RINGFR+9.0*(5.95E-3/PWBBP)+19.0*(1.87/(PWBBP*(PRES**.667)))+1.36*PWCIR+.224/PWBBI
2+7.25E-4/PWHFI
CALL OSWE (PWHFI,2,RINGWT)

```

TABLE XIV (CONTINUED)

```

CALL OSFR (PWHFI,.0133*PRES,6,2,RINGFR)
OWT=WT+RINGWT+(.232*(FLOW**1.5))/(PRES**.75)+.14*(PWHFI**3.0) +
1.011*(PWVHI**3.0)+(PWBBI**2.0)*(.0298*PWBBX+.15*PWBJY+.0477*PWBBI
2-.011*PWVHI)-.0967*PWJPY*(PWJPI**2.0)-.0436*PWBBI*(PWPGP**2.0)
REL=REL+RINGFR+.33/PWBBI
PWDS=TANA*((20.5*PWBBP**2.0)+(3.7*PWBBP**2.0))
PWDS1=17.2*PWDS
WT=WT+TOILW*PADS1
PUWT2=WT
PREL2=REL
PWLF=3175.0/PUMS
PINFT=48.0*PPPP4+19.0*PPPP5+S6*(6.0*PPPP4+4.0*PPPP5)
OPINFU=S6*(911.84+207.95*PWDS**0.5-515.22*PWDS+445.03*PWDS**1.5+
1400.00)+(1.0-S6)*(710.28+161.98*PWDS**0.5-401.33*PWDS+346.66*PWDS
2**1.5+380.0)
OPINFD=S6*((66000.0+18.0*PINFU)*PPPP4+88500.0*PPPP5)+(1.0-S6)*(
130000.0+18.0*PINFU)*PPPP4+69000.0*PPPP5)
GO TO 933
931 WT=0.0
REL=0.0
ANGL=0.0
PUMS=0.0
FLOW=0.0
PUWT2=0.0
PREL2=0.0
PWLF=0.0
PWDS1=0.0
PINFD=0.0
PINFT=0.0
PINFU=0.0
933 PWOP=0.0
IF (SENSE SWITCH 1) 921,922
921 PRINT 830,PRES,AMOM,WT,REL
922 PUNCH 830,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
PUNCH 830,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPR,VTEST,VTCST
PUNCH 830,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUHT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
PUNCH 830,PUWT1,PREL1,PAOP,PADS1,PFAFD,PFAFT
PUNCH 830,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFO
PUNCH 830,PINFT,PINFU
IF (X-10.0)205,205,206
205 X=X+.00001
GO TO 499
200 FORMAT (2E12.3)
201 FORMAT (5E10.3)
500 FORMAT (5E10.0)
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
206 STOP
END

```

TABLE XIV (CONTINUED)

```

IF (SENSE SWITCH 1)2050,499
2050 PRINT 2000
PRINT 2001
PRINT 2002
2000 FORMAT (46H OUTPUT FROM PROGRAM 5 HYDRAULIC INTENSIFIER)
2001 FORMAT (39H SYSTEM MOMENT INTENS. INTENS.)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
499 READ 820,PPPP1,ANGL1,ANGL2,PUMS1,PUMS2,S5,S6
READ 820,S7,S8,S9,PPPP2,PPPP3,PPPP4,PPPP5
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST, VTCST
READ 820,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLLX,XRUR1,XRUW1,VFAC3,PPPP9
READ 820,PUWT1,PREL1,PALE,PAOP,PADS1,PFAFD,PFAFT
READ 820,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
READ 820,PINFT,PINFU
X=1.0
IF(S9)861,861,862
862 PUMS=PUMS2
S6=0.0
ANGL=ANGL2
FLOW=ANUMB*ACTQA*PPPP9
TANA=SINF(ANGL)/COSF(ANGL)
PWBB1=1.0
5030 PWBBP=((FLOW/(PUMS*TANA*(20.5-1.92E-4*PRES)))-  

1(((PWBB1**2.0)*(3.58-3.46E-5*PRES))/(20.5-1.92E-4*PRES))+  

2((4.44E-3*PWBB1)/(20.5-1.92E-4*PRES))+  

3((1.64E-5*PRES*PWBB1)/((1.45*PWBB1+.26)*(TANA**2.0)*PUMS*  

4(20.5-1.92E-4*PRES))))**.333
IF (ABSF(PWBBP-PWBB1)-1.0E-5*PWBBP)501,501,502
502 PWBB1=(PWBBP+PWBB1)/2.0
GO TO 503
501 PWBB1=PWBBP*(3.98+2.46E-4*PRES)+.523
PWBBK=2.0*PWBBP
PWBBX=5.57*TANA*(2.9*PWBBP+.523)
PWBBJ=2.9*PWBBP+.523
PWBBR=(1.74E-5*PWBBP*TANA*(PRES**2.0))/FLOW
OWT=.185*((PWBB1**2.0)-.706*(PWBBP**2.0))*PWBBX-(95.5*(PWBBP**3.0)  

1-13.5*(PWBBP**2.0)-3.08*PWBBP+.438)*TANA-5.5*PWBBX*(PWBBP**2.0))
REL=PWBBR
CALL OSWE (PWBBK,2,RINGWT)
WT=WT+RINGWT*2.0
CALL OSFR (PWBBK,PRES,5,2,RINGFR)
REL=REL+RINGFR*2.0
WT=WT+9.0*(-176*PWBBX*PWBBP**2.0)
REL=REL+9.0*1.42E-2*PWBBL
CALL OSWE (PWBB1,2,RINGWT)
WT=WT+RINGWT*4.788
CALL OSFR (PWBB1,PRES,5,2,RINGFR)
REL=REL+RINGFR*2.0
WT=WT+(2.42*(PWBBP**3.0))+(1.62E-3*(PWBBP**3.0)*(PRES**.667))

```

TABLE XV

DECK 9- INTENSIFIER
COMPUTER PROGRAM LISTING

```

REL=REL+(2.36E-2/PWBBP)+(7.5/(PWBBP*(PRES**.667)))
PWVHI=PWBBI*(1.0+(1.516E-4*PRES))
OWT=WT+(.3*(FLOW**1.5)/(PRES**.75))
1+((PWBBI**3.0)*PRES*(2.06E-5+3.75E-9*PRES+2.2E-13*(PRES**2.0)))
PWHFI=.0911*(FLOW**.5)
WT=WT+2.34E-6*(FLOW**1.5)
REL=REL+.896/PWBBI+7.25E-4/PWHFI
CALL OSWE (PWHFI,2,RINGWT)
WT=WT+RINGWT
CALL OSFR (PWHFI,.0133*PRES,6,2,RINGFR)
REL=REL+RINGFR
OWT=WT+(.232*(FLOW**1.5)/(PRES**.75))+(.149*(PWHFI**3.0))
1+.022*((PWVHI**3.0))+((PWBBI**2.0)*(.0298*PWBBX-.022*PWVHI))
REL=REL+.33/PWBBI
PI DS=TANA*((20.5*PWBBP**2.0)+(3.7*PWBBP**2.0))
PI DS1=17.2*PI DS
WT=WT+TOILW*PI DS1
PREL3=REL
PUWT3=WT
PI LF=3220.0*PWBBP/(PUMS*TANA*(2.9*PWBBP+.523))
PTRFT=48.0*PPPP4+19.0*PPPP5+S6*(6.0*PPPP4+4.0*PPPP5)
OPTRFU=S6*(911.84+207.95*PI DS**0.5-515.22*PI DS+445.03*PI DS**1.5+
1400.00)+(1.0-S6)*(710.28+161.98*PI DS**0.5-401.33*PI DS+346.66*PI DS
2**1.5+380.0)
OPTRFD=S6*((66000.0+18.*PTRFU)*PPPP4+88500.0*PPPP5)+(1.0-S6)*(
130000.0+18.0*PTRFU)*PPPP4+69000.0*PPPP5)
GO TO 863
861 PUWT3=0.0
PREL3=0.0
PI LF=0.0
PI WP=0.0
PWOP=0.0
PI DS1=0.0
PTRFT=0.0
PTRFU=0.0
PTRFD=0.0
WT=0.0
REL=0.0
863 PUWT2=PUWT3+PUWT2
PREL2=PREL3+PREL2
PWLF=PI LF+PWLF
PINFT=PTRFT+PINFT
PINFD=PTRFD+PINFD
PINFU=PTRFU+PINFU
PWDS1=PWDS1+PI DS1
S8=S8+S9
PI WP=0.0
IF(SENSE SWITCH 1)851,852
851 PRINT 830,PRES,AMOM,WT,REL
852 PUNCH 830,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRDN
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3
PUNCH 830,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I

```

TABLE XV (CONTINUED)

```
PUNCH 830,PUWT1,PREL1,PALF,PAOP,PADS1,PFAFD,PFAFT
PUNCH 830,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
PUNCH 830,PINFU,PINFT,PPPP8,S7,S8,S9,PTRFD
PUNCH 830,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
IF (X-10.0)205,205,206
205 X=X+.00001
GO TO 499
401 FORMAT (2E12.3)
402 FORMAT (4E12.3)
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
206 STOP
END
```

```

*LIST
    IF (SENSE SWITCH 1)2050,960
2050 PRINT 2000
    PRINT 2001
    PRINT 2002
2000 FORMAT (32H OUTPUT FROM PROGRAM 6 FILTER)
2001 FORMAT (38H SYSTEM MOMENT FILTER FILTER)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
960 WEIGHT=0.0
RELIAB=0.0
READ 820,PPPP6,PPPP7,FFFF1,FFFF2,FFFF3,FFFF4,PPPP8
READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRRR1,RRRR3,RRRR4,QQQ1,QQQ2,QQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VpNUB,VHYSB,VWCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLLX,XRUR1,XRUW1,VFAC3
READ 820,FVOL,TWIGHT,TFAIL,TUCSU,TUCSD,TMT1I
READ 820,PUWT1,PREL1,PAOP,PADS1,PFAFD,PFAFT
READ 820,PFAFU,PUWT2,PREL2,PWL,PWOP,PWDS1,PINFO
READ 820,PINFT,PINFU,PPPP8,S7,S8,S9,PTRFD
READ 820,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
X=1.0
741 IF(FFFF1)931,931,932
932 FLOW=ACTQA*FFFF2
FOBOW=.05E-6*PRES*FLOW**1.5
FOBOR=.814E-4*FLOW/FOBOW
FOBOJ=.324*FLOW**0.5
FOBOI=FOBOJ+2.27E-5*FLOW**0.5*PRES
FOEOW=.0202*FLOW
FOWOW=FOBOJ**3.0*.0048
FOSOW=.00161*FOBOJ**2.33
CALL OSWE(FOBOI,1,RINGWT)
CALL OSFR(FOBOI,PRES,5,1,RINGFR)
FOHPW=RINGWT
FOHPR=RINGFR
FDBOW=.087*FOBOW
FDBOR=.008/FOBOJ
CALL OSWE(.623*FOBOJ,1,RINGWT)
CALL OSFR(.623*FOBOJ,20.,5,1,RINGFR)
RELIAB=RELIAB+RINGFR
WEIGHT=WEIGHT+RINGWT
CALL OSWE(.413*FOBOJ,1,RINGWT)
CALL OSFR(.413*FOBOJ,20.,5,1,RINGFR)
FDBLW=2.0*RINGWT
FDBLR=2.0*RINGFR
FDROW=0.0875*FDBOW
FIBOW=.0106+3.53E-6*PRES
FIBAW=.0004
CALL OSFR(.551,PRES,2,1,RINGFR)
RELIAB=RELIAB+RINGFR
ORELIAB=RELIAB+FOBOR+.1834*FOEOW+.012/FOBOJ+.0083/FOBOJ+.019/FOBOJ
1+FOBOR+FOHPR+.33*FOHPR+FDBOR+FDBLR*.167+FDBLR+.030/FOBOJ+FDBOR+
2.428*FDBOR+.0035+.00077/(FIBOW**.333)+.009+.030+.333*FIBAW+.030+
3.001+.002+.004+16.8/PRES+.008+.025+.009+.010

```

TABLE XVI

DECK NUMBER 10 - FILTER
COMPUTER PROGRAM LISTING

```

OWEIGHT=WEIGHT+FOBOW+FODEW+FOWOW+.225*FOWOW+FOSOW+2.12*FOBOW+FOHPW
1+1.125*FOHPW+FDBOW+FDBLW+FDBLW/2.0+3.2*FOSOW+FDROW+FDROW+FIBOW+
2FIBAW+.0001+.0003+FIBAW+.0003+.0038+.0017+.0008+4.0E-13*PRES**3.0+
3.0032+.0001+.0179+.0036
FIVOL=.095525*ACTQA*ANUMB**1.5-.07071*ACTQA*ANUMB
FWGHT=WEIGHT+FIVOL*TOILW
FFAIL=RELIAB
FPURT=TMT1I*(FFFF2*ANUMB/2.0)**0.5
FUCSU=(56.+280.*FPURT+9.1/FPURT**2.0)*FFFF1
FUCSD=(15000.+10.*FUCSU)*FFFF3+42000.*FFFF4
GO TO 933
931 FUCSD=0.0
FUCSU=0.0
FIVOL=0.0
FFAIL=0.0
FWGHT=0.0
933 IF (SENSE SWITCH1) 821,822
821 PRINT 830,PRES,AMOM,FWGHT,FFAIL
822 PUNCH 830,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
PUNCH 830,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLX,XRUR1,XRUW1,VFAC3
PUNCH 830,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
PUNCH 830,PUWT1,PREL1,PAWF,PAOP,PADS1,PFAFD,PFAFT
PUNCH 830,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
PUNCH 830,PINFT,PINFU,PPPP8,S7,S8,S9,PTRFD
PUNCH 830,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
PUNCH 830,FWGHT,FFAIL,FIVOL,FUCSU,FUCSD,FFFF1,PPPP8
IF (X-3.0)950,950,951
950 X=X+.000001
GO TO 960
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
951 STOP
END

```

```

IF (SENSE SWITCH 1) 2050,730
2050 PRINT 2000
PRINT 2002
2000 FORMAT (46H OUTPUT FROM PROGRAM 7 RESERVOIR-ACCUMULATOR)
2002 FORMAT (40H PRESSURE ARM WEIGHT FAIL RATE)
730 READ 820,SSS2,SSS3,SSSI,RSPA1,RSPA2,RSPA3,TOILW
READ 820,RRR1,RRR3,RRR4,QQQ1,QQQ2,QQQ3,VNAFQ
READ 820,VHSFO,VFLRC,VPNUB,VHYSB,VNCST,VDEVL,VPEND
READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
READ 820,ALLX,XRUR1,XRURW1,VFAC3
READ 820,FVOL,TWHT,TFAIL,TUCSU,TUCSD,TMT1I
READ 820,PUWT1,PREL1,PALE,PAOP,PADS1,PFAFD,PFAFT
READ 820,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
READ 820,PINFU,PINFU,PPPP8,S7,S8,S9,PTRF0
READ 820,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
READ 820,FWGHT,FFAIL,FIVOL,FUCSU,FUCSD,FFFF1,PPPP8
IF (S7+S8+S9) 971,971,972
971 RAWGT=0.0
RAFAL=0.0
RAPCU=0.0
RAPCD=0.0
RSPA3=0.0
RSPA2=0.0
RPAPI=0.0
RSPA1=0.0
RPRE=40.0
RPARI=0.0
RHPP1=0.0
SACSU=0.0
GO TO 822
972 RPRE=SSS2*PRES+SSS3
SVOL=ACTQM*ACTIM*ANUMB*RRR1
SVOLW=SVOL*TOILW
ORVOL=SVOL+(0.20+1.667E-6)*(SVOL+FVOL+(ANUMB*ACVOL)+FIVOL+PADS1+
1PWDS1+PIDS1)
RVOLW=RVOL*TOILW
RPAPI=0.0
720 RPAPIT=(2.5464*RVOL+RPAPI*.32490)**.333
IF (ABSF(RPAPIT-RPAPI)-.000001*RPAPI) 722,722,721
721 RPAPI=(RPAPIT+RPAPI)/2.
GO TO 720
722 RPARI=.1365*RPAPI+(4.550E-4*RPRE*RPAPI**2.)/(RPAPI-2.92)
RPAPW=.0093*(RPAPI)**2.+6.80E-6*RPRE*RPAPI**4++0.046
RPARW=6.550E-5*(RPAPI+7.)*(RPAPI)**3.*RPRE/(RPAPI-2.9)
RHPP1=((RPAPI**2.-.3249)*RPRE/PRES+RPARI**2.)**.5
CALL OSWE(RPAPI,2,RINGWT)
WEIGHT=0.0017+RINGWT*1.872+.0028+1.3750E-5*RPRE*RPAPI**2.+0.0004
ORELIAB=7.49/((RPRE*RPAPI)**2.)+1.070E-4*RPAPI**2.+2.50E-4*RPRE*
1RPAPI**2.
CALL OSFR(RPAPI,RPRE,1,2,RINGFR)
RELIAB=RELIAB+RINGFR*1.0666
CALL OSWE(RHPP1,2,RINGWT)
WEIGHT=WEIGHT+RINGWT
CALL OSFR(RHPP1,RPRE,3,1,RINGFR)

```

TABLE XVII

DECK NUMBER 11 - RESERVOIR-ACCUMULATOR
COMPUTER PROGRAM LISTING

```

RELIAB=RELIAB+RINGFR
CALL OSFR(1.50,RPRE,5,1,RINGFR)
RELIAB=RELIAB+RINGFR
CALL OSFR(0.55,RPRE,3,1,RINGFR)
RELIAB=RELIAB+RINGFR*1.5
SPAPI=0.0
IF (SSSI) 535,535,536
536 SPAGI=RHPP1+.224
723 SPAPIT=(6.366*SVOL+SPAPI*SPAGI**2.0)**.333
IF (ABSF(SPAPIT-SPAPI)-.000001*SPAPI) 725,725,724
724 SPAPI=(SPAPIT+SPAPI)/2.
GO TO 723
7250 SPAPW=(.012184*SPAPI**3.0+.04383*SPAGI**2.0*SPAPI+(3.52860E-7*
1PRES*(SPAPI**2.0-SPAGI**2.0)**2./SPAGI))*SSSI
SPAPR=(7.0990E+3*SPAGI/(PRES*(SPAPI**2.-SPAGI**2.0)**2.))*SSSI
SPAGW=0.129*RPAPI*(RHPP1+.112)*SSSI
SPAGR=2.5320E-4*(RHPP1*RPAPI+SPAGI*SPAPI)*SSSI
SPGNW=0.093*SPAGI*SSSI
SPGNR=(.0015/SPAGI)*SSSI
CALL OSWE(SPAPI,2,RINGWT)
WEIGHT=WEIGHT+RINGWT*3.744
CALL OSFR(SPAPI,PRES,1,2,RINGFR)
RELIAB=RELIAB+RINGFR*3.2*SSSI
CALL OSWE(RPARI,1,RINGWT)
WEIGHT=WEIGHT+RINGWT*1.68
CALL OSFR(RPARI,PRES,3,1,RINGFR)
RELIAB=RELIAB+RINGFR*1.05*SSSI
CALL OSWE(SPAGI,1,RINGWT)
WEIGHT=WEIGHT+RINGWT*3.778
CALL OSFR(SPAGI,PRES,3,1,RINGFR)
RELIAB=RELIAB+RINGFR*2.2*SSSI
SCACI=SPAPI+3.1020E-5*PRES*SPAPI
OSHPCW=0.44+(9.678E-6*SPAPI**2.+4.6624E-6*(SPAPI**2.-SPAGI**2.))**
1PRES*SPAPI*SSSI
SHPCR=3.4652E-5*SPAPI*SSSI
CALL OSWE(SCACI,2,RINGWT)
WEIGHT=WEIGHT+(RINGWT*1.0541+.0106+.0003+.0037*0.649)*SSSI
CALL OSFR(SCACI,PRES,2,2,RINGFR)
RELIAB=RELIAB+(RINGFR*1.333+.010)*SSSI
CALL OSFR(0.50,PRES,5,1,RINGFR)
RELIAB=RELIAB+RINGFR*SSSI
CALL OSFR(1.50,PRES,5,1,RINGFR)
RELIAB=RELIAB+RINGFR*1.333*SSSI
ORHXXW=4.71620E-6*RPRE*RPAPI**3.*(.+0.007662*RPRE)+(2.1860E-6*
1PRES*SPAPI*(SPAPI**2.-SPAGI**2.))*(SSSI)+(2.5970E-7*((1.256*SPAPI
2-2.125)**2.-2.1**2.)*PRES/(1.256*SPAPI-2.125))+.0848+.46383*SPAPI
GO TO 537
535 SPAGI=0.0
SPAPW=0.0
SPAGW=0.0
SPGNW=0.0
SHPCW=0.0
SACSU=0.0
SPAGR=0.0
SPGNR=0.0
SHPCR=0.0
SPAPR=0.0
537 RCASW=1.7220E-4*RPAPI**3.0*RPRE
RCASR=2.69/(RPAPI*RPRE)

```

```

CALL OSWE(RPAPI,2,RINGWT)
WEIGHT=WEIGHT+RINGWT*1.748+.0925+.0006
CALL OSFR(RPAPI,RPRE,2,2,RINGFR)
RELIAB=RELIAB+RINGFR*1.333+.0001*RPRE
CALL OSFR(.644,RPRE,5,1,RINGFR)
RELIAB=RELIAB+RINGFR
RHXXW=4.7162E-6*RPRE*RPAPI**3.0*(1.0+.007662*RPRE)
RHXXR=1.17E-2*RPAPI**2.
WEIGHT=WEIGHT+.0045+.0002+.0024*SSSI+.0925*SSSI+.0587+.1432
CALL OSFR(1.0625,RPRE,5,1,RINGFR)
RELIAB=RELIAB+RINGFR*.3+.010
OSH_RXW=(732.173*(SPAPI+3.1020E-5*PRES*SPAPI)*(2.7010E-6*(SPAPI
1**2.-SPAGI**2.))+.003)*SSSI
RELIAB=RELIAB+.007*SSSI+.004*SSSI
CALL OSFR(.75,PRES,5,1,RINGFR)
RELIAB=RELIAB+RINGFR*.4.*SSSI+.013+.005
OWEIGHT=WEIGHT+(.050+.0148+.026+.2213+.2744+.0015+.0025+.1986+.0142
1+.4737*(1.-(1.-.2.0660E-5*PRES*2.))**2.)*SSSI+.252*RSPA3+.3138*
2RPAPI*RSPA3
ORELIAB=RELIAB+.012+.005+.018+.003+.004+.015+.004+.003+.062+.030
1*SSSI+RSPA3*(.0968+RSPA1*.250+RSPA2*.150)+RSPA3*(RPAPI/2.)*(.152+
2RSPA1*.242+RSPA2*.196)
OWEIGHT=WEIGHT+RPAW+RPARH+SPAPW+SPAGW+SPGNW+SHPCW+RCASW+RHXXW+
1SH_RXW
RELIAB=RELIAB+SPAPR+SPAGR+SPGNR+SHPCR+RCASR+RHXXR
RAWGT=WEIGHT
TUCSU=TUCSU+(140.0+4.6*RAWGT**0.5)*(1.0+.25*SSSI)
RAFAL=RELIAB
OSACSU=(2.7492*(71.82*SPAGI-9.77*SPAGI**2.0+2.014*SPAGI**3.+SPAPI**
13.0*(.875758-200.9068/PRES-79711.07/PRES**2.0)+66628232.0/PRES**
22.0-117434.89/PRES+105.55801-.034223256*PRES+1.0761896E-5*PRES**
32.0)*(1.0+.0188*(SPAGI/5.))+50.)*SSSI
IF (SENSE SWITCH 1)821,822
821 PRINT 830,PRES,AMOM,RAWGT,RAFAL
822 PUNCH 830,RRR1,RRRR3,RRRR4,QQQQ1,QQQ02,QQQ03,VNAFQ
PUNCH 830,VHSFO,VFLRC,VPNUB,VHYSB,VVCST,VDEVL,VPEND
PUNCH 830,VREPR,VLIFF,VLIFA,VCYCA,VOPER,VTEST,VTCST
PUNCH 830,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
PUNCH 830,ACTWT,ACTRB,ANAFO,ALIFE,ACYCL,ADCST,AUCST
PUNCH 830,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
PUNCH 830,ALLLX,XRUR1,XRUW1,VFAC3
PUNCH 830,FVOL,TWGBT,TFAIL,TUCSU,TUCSD,TMT1I
PUNCH 830,PUWT1,PREL1,PALF,PAOP,PADS1,PFAFD,PFAFT
PUNCH 830,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
PUNCH 830,PINFT,PINFU,PPPP8,S7,S8,S9,PTRFD
PUNCH 830,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
PUNCH 830,FWGHT,FFAIL,FIVOL,FUCSU,FUCSD,FFFF1,PPPP8
PUNCH 830,RAWGT,RAFAL,RSPA3,RSPA2,RPAPI,RSPA1,RPRE
PUNCH 830,RPARI,RHPI,SSSI,SACSU
. X=1.0
. IF (2.0-X)745,730,730
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
745 STOP
END

```

```

X=1.0
554 READ 820,RRRR1,RRRR3,RRRR4,QQQQ1,QQQQ2,QQQQ3,VNAFQ
      READ 820,VHSFD,VFLRC,VPNUB,VHYSB,VWCST,VDEVL,VPEND
      READ 820,VREPR,VLIFP,VLIFA,VCYCA,VOPER,VTEST,VTCST
      READ 820,PRES,AMOM,ACTIM,XRUSU,ACTQM,ACTQA,ACTQL
      READ 820,ACTWT,ACTRB,ANAFQ,ALIFE,ACYCL,ADCST,AUCST
      READ 820,ACVOL,ADTIM,XRUWT,XRURB,XRUSD,ANUMB,VPRON
      READ 820,ALLX,XRUR1,XRUW1,VFAC3
      READ 820,FVOL,TWGHT,TFAIL,TUCSU,TUCSD,TMT1I
      READ 820,PUWT1,PREL1,PALF,PAOP,PADS1,PFAFD,PFAFT
      READ 820,PFAFU,PUWT2,PREL2,PWLF,PWOP,PWDS1,PINFD
      READ 820,PINFT,PINFU,PPPP8,S7,S8,S9,PTRFD
      READ 820,PUWT3,PREL3,PILF,PIWP,PIDS1,PTRFT,PTRFU
      READ 820,FWGHT,FFAIL,FIVOL,FUCSU,FUCSD,FFFF1,PPPP8
      READ 820,RAWGT,RAFAL,RSPA3,RSPA2,RPAPI,RSPA1,RPRE
      READ 820,RPARI,RHPP1,SSSI,SACSU
      VLIFA=.1+VLIFA
      VCYCL=.1+VCYCL
      VLIFP=.1+VLIFP
      IF (RPARI) 181,181,182
181 RSPAU=0.0
      REC SU=0.0
      GO TO 183
182 RSPAU=RSPA3*(281.30+32.30*RSPA2+40.10*RPAPI+15.60*RPAPI*(RSPA1+
      1RSPA2))
      OREC SU=2.7492*(22.3786+RPAPI**3.0*(.875758-200.9068/(RPRE+500.-
      179711.07/(RPRE+500.)*2.0)+66628232.0/(RPRE+500.0)**2.0-117434.89/
      2(RPRE+500.)*105.55801-.034223256*(RPRE+500.)*1.0761896E-5*(RPRE+500.)*2.0)*(1.0+.0188*(RPAPI/2.))
      OREC SU=RECSU+2.7492*(71.82*RPARI-9.77*RPARI**2.0+2.014*RPARI**3.0+
      1RHPP1**3.0*(.875758-200.9068/PRES-79711.07/PRES**2.0)+66628232.0/
      2PRES**2.0-117434.89/PRES+105.55801-.034223256*PRES+1.0761896E-5*
      3PRES**2.0)*(1.0+.0188*(RPAPI/2.))+520.
183 RAPCU=RECSU+SACSU+RSPAU
      ORAPCD=(74000.+20000.*SSSI+RAPCU*10.)*RRRR3+(69000.+5700.*SSSI)*
      1RRRR4
      OTUCSU=TUCSU+(140.0+4.6*PUWT1**0.5)*S7+(140.+4.6*PUWT2**0.5)*S8+(
      1110.+1.15*FWGHT**0.5)*FFFF1+(140.+4.6*RAWGT**0.5)*(1.0+.25*SSSI)
      QWGBT=1.84E-6*PRES**2.0*TMT1I**3.0*(ANUMB*QQQQ3/2.0)**1.5
      QPORT=TMT1I*(ANUMB*QQQQ3/2.0)**0.5
      CALL OSFR(QPORT,PRES,3,1,RINGFR)
      QFAIL=RINGFR+0.20
      CALL OSFR(1.1*QPORT,PRES,5,1,RINGFR)
      QFAIL=QFAIL+RINGFR
      CALL OSFR(1.1*QPORT,40.0,7,1,RINGFR)
      QFAIL=QFAIL+RINGFR
      CALL OSFR(1.35*QPORT,40.0,3,1,RINGFR)
      QFAIL=QFAIL+RINGFR
      CALL OSFR(1.35*QPORT,40.0,5,1,RINGFR)
      QFAIL=QFAIL+RINGFR
      QDCSU=68.55*(TMT1I*(QQQQ3*ANUMB/2.0)**.5)**2*26286+532*53+300.
      QDCSD=(20000.+10.*QDCSU)*QQQQ1+42000.*QQQQ2
      YTPRW=.0431
      YTPRR=.0628
      YTPRF=YTPRR
      YTPRU=250.0
      OVHYSW=(ACTWT+XRUWT)*ANUMB+TWGHT*ANUMB/2.0+PUWT1*S7+PUWT2*S8+RAWGT

```

TABLE XVIII
DECK NUMBER 12 - OVERALL VEHICLE
COMPUTER PROGRAM LISTING

```

1+FWGHT*FFFF1+YTPRW+QWGHT
OVHYSR=(ACTRB+XRURB)*ANUMB+TFAIL*ANUMB/2.0+PREL1*S7+PREL2*S8+RAFAL
1+FFAIL*FFFF1+YTPRF+QFAIL
VHSRC=.99*VHYSR*VHYSB*VHSFO*VFLRC*1.0E-6
VHSC=VHYSW*VHYSB*VPNUB*VWCST
IF (PPPP8-1.0) 301,302,303
301 VPMCLC=1.0E6
VPUC=0.0
S10=0.0
GO TO 304
302 VPMCLC=PALF
VPUC=PFAFU
S10=S7
GO TO 304
303 VPMCLC=PWL
VPUC=PINFU
S10=S8
304 IF (ALIFE/VLIFA-ACYCL/VCYCA) 306,306,305
305 VQUAM=ACYCL/VCYCA
GO TO 308
306 VQUAM=ALIFE/VLIFA
308 IF (VQUAM-1.0) 310,310,309
309 VQUAM=1.0
3100VHSLC=ANUMB*VHYSB*VPNUB*AUCST*VREPR*(VQUAM-1.0)+VHYSB*VPNUB*VPUC*
1(VLIFP/VPMCLC-1.0)*VREPR*S10
0VPHDU=ADCS+PFAFD+PINFD+FUCSD+TUCSD+QDCSD+XRSUD+RAPCD+((AUCST+
1XRSU)*(ANUMB)+PFAFU*S7+PINFU*S8+FUCSU+TUCSU+QDCSU+RAPCU+YTPRU)
2*VHYSB*VPNUB
IF(PFAFT-PINFT)400,400,401
400 VHCDT=PINFT
GO TO 403
401 VHCDT=PFAFT
403 IF(VHCDT-ADTIM)404,404,405
404 VHCDT=ADTIM
405 IF (VHCDT-VDEVL)406,406,407
406 VHCDT=VDEVL
407 VHDT=VHCDT-VDEVL)*VPEND
PAOP=1.35E-3*ACTQL*ANUMB*PRES/VHYSW
PWOP=1.26E-3*ACTQL*ANUMB*PRES/VHYSW
PIWP=1.00E-3*ACTQL*ANUMB*PRES/VHYSW
PWOP=PIWP+PWOP
IF (PWOP-PAOP)313,314,314
313 PMOP=PAOP
GO TO 315
314 PMOP=PWOP
IF(PMOP)316,316,315
316 VHOTC=0.0
GO TO 318
315 VHOTC=VTEST*VHYSB*VPNUB*(VOPER/(PMOP-1.0)*VTCST)
318 VPCST=VHSRC+VHSC+VHSLC+VPHDU+VHDT+VHOTC
PRINT 900,VPRON
PRINT 901,PRES,AMOM
PRINT 911,VPCST
PRINT 902,VHYSW
PRINT 903,VHYSR
PRINT 904
PRINT 912
PRINT 905
PRINT 906,ACTWT,ACTRB,ALIFE,ACYCL

```

```
PRINT 907,PUWT1,PREL1,PALF
PRINT 908,PUWT2,PREL2,PWL
PRINT 913
PRINT 909,XRUWT,XRURB
PRINT 910,RAWGT,RAFAL
PRINT 912
PRINT 912
900 FORMAT (13X,24H COMPUTER PROGRAM NUMBER,F10.0)
901 FORMAT (15X,F10.0,4H PSI,9X,F10.0,16H INCH MOMENT ARM)
911 FORMAT (21H TOTAL SYSTEM COST IS,F11.0,8H DOLLARS)
902 FORMAT (23H TOTAL SYSTEM WEIGHT IS,F10.1,7H POUNDS)
903 FORMAT (37H TOTAL SYSTEM GENERIC FAILURE RATE IS,F10.2)
904 FORMAT (10X,27H FAILURES PER MILLION HOURS)
905 FORMAT (20X,50H WEIGHT FAILURE RATE LIFE-HOURS LIFE-CYCLES)
906 FORMAT (18H ACTUATOR ,F9.0,F14.2,F13.2,F15.0)
907 FORMAT (18H FIXED ANGLE PUMP ,F9.0,F14.2,F13.2)
908 FORMAT (18H WOBBLE PLATE PUMP,F9.0,F14.2,F13.2)
913 FORMAT (3X,15H DR INTENSIFIER)
909 FORMAT (18H TRUSS ,F9.0,F14.2)
910 FORMAT (18H RESERVOIR ,F9.0,F14.2)
912 FORMAT(20X)
820 FORMAT (7E10.0)
830 FORMAT (7E10.3)
840 FORMAT (10X,6E10.3)
IF (X) 553,553,554
553 STOP
END
```